



Universitat Oberta de Catalunya

**PFC**

**Construcció d'un SIG per la gestió de rutes en  
camins no cartografiats**

**Olga Gualdo Sesma**

**Enginyeria Informàtica (2n cicle)**

**Curs 2006/2007 (1er Semestre)**

**Consultor: Eduard Allué Pont**



Universitat Oberta de Catalunya

A Jordi, pel seu suport en les meves hores baixes i paciència en les meves llargues  
jornades davant de l'ordinador. I a la meva família.

## Resum

Aquest projecte té la finalitat de crear un sistema d'informació geogràfica (SIG), que permeti gestionar les rutes que realitzen els veterinaris d'un laboratori d'analítiques quan visiten els seus clients en les granges. També té que planificar recorreguts a diversos destins utilitzant la informació emmagatzemada.

Per tal de resoldre aquest objectiu es fa un estudi de quines són les característiques d'un SIG, així com quins conceptes del món de la cartografia i la geodèsia són necessaris per portar-lo a terme. Per altra banda, també es fa un estudi dels diversos algorismes de recerca de camins que ajudaran a poder planificar els recorreguts dels veterinaris.

Seguidament es construirà un SIG amb el programari comercial Geomedia Professional 6.0 d'Intergraph, el qual permetrà visualitzar la informació geogràfica que es té de la zona d'acció del laboratori, en aquest cas Catalunya, i la informació de les rutes emmagatzemades. Mitjançant Microsoft Visual Basic 6.0 es interaccionarà amb Geomedia Professionals 6.0 per crear comandes que ajudin al supervisor del laboratori a gestionar i planificar les rutes.

Aquestes comandes permetran que el supervisor pugui afegir noves rutes a partir d'un fitxer, cercar rutes pel seu origen o destí, cercar rutes per la data de realització d'aquestes, calcular els punt d'interseccions entre rutes i per finalitzar pugui planificar recorreguts des de la central del laboratori a les granges dels clients.

**ÍNDEX**

ÍNDEX TAULES .....	6
ÍNDEX FIGURES .....	7
1. INTRODUCCIÓ .....	8
1.1. Justificació del PFC .....	8
1.2. Objectius .....	8
1.3. Enfocament i mètode seguit .....	9
1.4. Planificació .....	9
1.5. Productes Obtinguts .....	10
1.6. Descripció resta capítols .....	11
2. INTRODUCCIÓ AL SIG .....	12
2.1. Característiques principals d'un SIG .....	12
2.2. Components d'un SIG .....	12
2.2.1. Tipus de dades .....	13
2.2.2. Conjunt de dades .....	13
2.2.3. Models de representació de dades geogràfiques .....	14
2.2.3.1. Model Vectorial .....	14
2.2.3.2. Model <i>raster</i> .....	15
2.3. Formats de fitxers .....	15
2.3.1. Formats de fitxers pel model vectorial .....	15
2.3.2. Formats de fitxers pel model raster .....	16
3. CONCEPTES CARTOGRÀFICS I GEODÈSICS .....	17
3.1. Definició de cartografia .....	17
3.1.1. Coordenades geogràfiques .....	17
3.2. Definició de geodèsia .....	18
3.2.1. Geoide, el·lipsoide i <i>datum</i> .....	19
3.3. Projeccions cartogràfiques .....	20
3.3.1. Projeccions polars .....	20
3.3.2. Projeccions còniques .....	21
3.3.3. Projeccions cilíndriques .....	22
3.3.4. Projecció UTM .....	22
3.3.4.1. Coordenades UTM .....	23
4. GEOMEDIA PROFESSIONAL 6.0 .....	25
4.1. Introducció a Geomedia Professional 6.0 .....	25
4.2. Elements principals de Geomedia Professional 6.0 .....	26
4.3. Utilització de Geomedia Professional 6.0 .....	26
4.3.1. Connexió a les dades .....	26
4.3.2. Sistema de coordenades .....	27
4.3.3. Mapes i dades .....	28
4.3.4. Entitats .....	28
4.3.5. Captura i correcció de dades .....	29
4.3.6. Anàlisis .....	29
4.4. Interacció amb Visual Basic 6.0 .....	29
5. ALGORITMES DE RECERCA DE CAMINS .....	31
5.1. Teoria de Grafs .....	31
5.2. Tipus d'algoritmes de recerca de camins .....	32

5.2.1.	Algoritme Profunditat - Primer .....	32
5.2.2.	Algoritme Amplada - Primer.....	33
5.2.3.	Algoritme A* .....	34
5.3.	Algoritme escollit .....	36
6.	CREACIÓ D'UN SIG.....	38
6.1.	Disseny i creació d'un SIG en l'àmbit de Catalunya .....	38
6.1.1.	L'ICC .....	38
6.1.2.	Finalitats de l'ICC.....	38
6.1.3.	Catàleg de l'ICC .....	39
6.1.4.	Crear un <i>geoworkspace</i> .....	40
6.1.5.	Importació de les dades de l'ICC .....	40
6.1.6.	Ortofotos de la zona de Lleida.....	44
6.2.	Model de Dades .....	46
6.3.	Incorporació d'entitats en el <i>geoworkspace</i> .....	48
7.	IMPLEMENTACIÓ DE COMANDES .....	49
7.1.	Requeriments .....	49
7.2.	Anàlisis .....	51
7.2.1.	Actors .....	51
7.2.2.	Comandes.....	51
7.2.2.1.	Comanda "Entrada d'una nova Ruta" .....	52
7.2.2.1.1.	Resum .....	52
7.2.2.1.2.	Funcionament.....	53
7.2.2.1.3.	Codi Font.....	55
7.2.2.2.	Comanda "Recuperació de recorreguts per data" .....	58
7.2.2.2.1.	Resum .....	58
7.2.2.2.2.	Funcionament.....	59
7.2.2.2.3.	Codi.....	60
7.2.2.3.	Comanda "Recuperació de rutes per origen i destí" .....	62
7.2.2.3.1.	Resum .....	62
7.2.2.3.2.	Funcionament.....	62
7.2.2.3.3.	Codi.....	63
7.2.2.4.	Comanda "Introducció de les granges conegudes" .....	64
7.2.2.4.1.	Resum .....	64
7.2.2.4.2.	Funcionament.....	64
7.2.2.4.3.	Codi.....	65
7.2.2.5.	Comanda "Identificació de creuaments" .....	66
7.2.2.5.1.	Resum .....	67
7.2.2.5.2.	Funcionament.....	68
7.2.2.5.3.	Codi.....	69
7.2.2.6.	Comanda "Planificació de rutes" .....	72
7.2.2.6.1.	Resum .....	73
7.2.2.6.2.	Funcionament.....	74
7.2.2.6.3.	Codi.....	77
8.	CONCLUSIONS .....	86
8.1.	Noves línies de treball.....	87
9.	BIBLIOGRAFIA .....	88

**ÍNDIX TAULES**

Taula 1 Planificació projecte .....	10
Taula 2 Avantatges i desavantatges del model vectorial.....	15
Taula 3 Avantatges i desavantatges del model <i>raster</i> .....	15
Taula 4 Taula amb las característiques del <i>geoworkspace</i> .....	40
Taula 5 Taula amb el paràmetres de l'arxiu de coordenades .....	40
Taula 6 Taula de propietats de les entitats de la base municipal de Catalunya .....	42
Taula 7 Descripció de les escales utilitzades per cada entitat .....	43
Taula 8 Taula del càlcul de coordenades de la imatge.....	45
Taula 9 Descripció entitats sistema.....	48
Taula 10 Coordenades Central.....	50
Taula 11 Resum de la comanda "Entrada d'una nova Ruta" .....	52
Taula 12 Resum de la comanda "Recuperar per Data" .....	59
Taula 13 Resum de la comanda "Recuperar per Origen i Destí" .....	62
Taula 14 Resum de la comanda "Incorporar Granges" .....	64
Taula 15 Resum de la comanda "Identificació de creuaments".....	67
Taula 16 Resum de la comanda "Planificació de Rutes" .....	74

**ÍNDIX FIGURES**

Figura 1 Representació meridians i paral·lels [9] ..... 18

Figura 2 Relació entre geoide, el·lipsoide i *datum* [17] ..... 19

Figura 3 Projecció polar des de l'hemisferi nord [13] ..... 21

Figura 4 Projecció cònica ..... 21

Figura 5 Projecció UTM[24]..... 22

Figura 6 Coordenades UTM..... 24

Figura 7 Llistat de connexions actives a Geomedia Professional 6.0..... 27

Figura 8 Finestra de visualització de dades ..... 28

Figura 9 Representació d'un graf ..... 31

Figura 10 Estat inicial de l'algoritme profunditat - primer i els seus possibles moviments . 32

Figura 11 Segon estat i els seus possibles moviments ..... 33

Figura 12 Selecció de nodes de l'algoritme d'amplada - primer ..... 34

Figura 13 Estat inicial algoritme A\* ..... 35

Figura 14 Valors de la funció heurística dins del graf ..... 35

Figura 15 Pantalla de la utilitat "Definir archivo de sistema de coordenadas" ..... 41

Figura 16 Pantalla de canvi de l'algoritme de projecció..... 41

Figura 17 Dades de la connexió a un fitxer CAD ..... 42

Figura 18 Visualització a escala 1:111.853.762 ..... 43

Figura 19 Visualització a escala 1:13.981.720..... 44

Figura 20 Ortofoto inserida dins del SIG..... 45

Figura 21 Diagrama del model de dades en UML..... 46

Figura 22 Definició d'entitats ..... 48

Figura 23 Pantalla de la comanda "Entrada d'una nova Ruta" ..... 53

Figura 24 Funcionament de la comanda "Entrada d'una nova Ruta" ..... 54

Figura 25 Pantalla de la comanda "Recuperació de recorreguts per data" . ..... 60

Figura 26 Pantalla i funcionament de la comanda "Recuperació de rutes per Origen i Destí"  
..... 63

Figura 27 Pantalla de la comanda "Introducció de les granges conegudes" ..... 65

Figura 28 Tipus de creuament entre rutes ..... 66

Figura 29 Pantalla "Identificació de creuaments" ..... 68

Figura 30 Missatge de finalització de la comanda "Identificació de creuaments" ..... 68

Figura 31 Representació gràfica del mètode per omplir la LNP..... 73

Figura 32 Pantalla de la comanda "Planificació de rutes" ..... 75

Figura 33 Selecció de destins en la comanda "Planificació de rutes" ..... 75

Figura 34 Primer destí trobat. .... 76

Figura 35 Segon i tercer destins trobats ..... 76

Figura 36 Descripció de les rutes trobades per cada destí..... 77

---

---

## 1. INTRODUCCIÓ

---

---

En aquest capítol es presenta el projecte de final de carrera per la construcció d'un SIG que permeti gestionar les rutes realitzades per veterinaris en les seves visites a granges per camins no cartografiats.

En el primer apartat es justifica el motiu d'aquest projecte, a continuació es llista quins són els objectius del mateix. Seguidament s'explica quin ha estat el mètode a seguir per resoldre aquest projecte i quina ha estat la seva planificació.

Finalment es fa una breu descripció del producte obtingut, així com una explicació de la resta de capítols de la memòria.

### 1.1. Justificació del PFC

Aquest projecte planteja el problema que té una empresa de sanitat animal, quan a l'hora de visitar les diverses granges es troben que els camins, des de les carreteres convencionals a les granges, no estan cartografiats.

Per això es proposa la creació d'un SIG que ajudi a gestionar aquestes rutes que els veterinaris emmagatzemen la primera vegada que hi van, de manera que el següent cop que hi vagin, serà molt més fàcil i ràpid arribar-hi. Per altra banda es planteja la necessitat de gestionar la manera de trobar un camí per recorre diverses granges d'una sola vegada, partint de les rutes emmagatzemades fins aquell moment.

### 1.2. Objectius

Els objectius que es volen assolir en aquest projecte són els següents:

- ✓ Conèixer les característiques principals d'un SIG, quins components ho formen, així com quines són les aplicacions que tenen.



- ✓ Conèixer el programari Geomedia Professional 6.0 d'Intergraph.
- ✓ Construir un SIG amb Geomedia Professional 6.0 i programar comandes amb Microsoft Visual Basic 6.0.
- ✓ Conèixer alguns tipus d'algoritmes de recerca de camins, per tal de resoldre les funcionalitats que planteja l'enunciat.

### 1.3. Enfocament i mètode seguit

L'enfocament i el mètode seguit en aquest projecte ha estat el següent:

1. Seleccionar les fonts d'informació. Una de les tasques més importants ha estat trobar unes bones fonts d'informació, per això s'ha fet servir internet en la majoria de casos, així com llibres que ja havien ajudat a altres projectistes a desenvolupar els seus projectes.
2. Redacció de la memòria. La memòria del projecte és una de les parts més important i, és per això, que s'han utilitzat moltes de les hores d'aquest projecte a fer un bon redactat. Per aconseguir-ho, primer s'abocaven totes les idees sobre el text i, després punt per punt, s'anava polint parant atenció a la redacció. Per aquesta part ha estat molt important l'ajuda del consultor, que amb les seves revisions ha fet que l'expressió del redactat hagi anat millorant en els transcurso del projecte.
3. Realització de la part pràctica. Per la realització de la part pràctica s'ha utilitzat el gran nombre d'exemples que l'ajuda de Geomedia Professional 6.0 té, així com altres projectes ja realitzats. Una vegada s'ha implementat la solució, s'ha documentat en la memòria de manera pertinent, afegint captures d'imatge i taules explicatives.

### 1.4. Planificació

A continuació en la taula 1 es mostrarà un resum de la planificació a seguir durant el projecte:

<i>Activitat</i>	<i>Data inici</i>	<i>Data fi</i>
<u>Creació del pla de treball</u> . Document que serveix per planificar i gestionar el projecte.	20/09/2006	02/10/2006
<u>Introducció als SIG</u> . Estudi de les característiques d'un SIG.	03/10/2006	24/10/2006
<u>Introducció al Geomedia Professional 6.0</u> . Estudi del programari Geomedia Professional 6.0 per tal de poder crear el SIG del projecte en aquest producte comercial.	24/10/2006	08/11/2006
<u>Definició d'un SIG</u> . Crear un SIG a l'àmbit del Catalunya i definir el model lògic i físic de dades per resoldre el projecte.	09/11/2006	23/11/2006
<u>Creació del Programari Gestió de Rutes</u> . Creació de les comandes necessàries per tal de resoldre l'enunciat.	24/11/2006	25/12/2006
<u>Revisió de la memòria</u> . Activitat per resoldre els diferents errors que hi puguin haver en la memòria.	25/12/2006	30/12/2006
<u>Realització de la presentació</u> . Generar un document amb el programari Microsoft Power Point, amb la presentació del projecte.	31/12/2006	05/01/2007
<u>Lliurament del projecte</u> . Lliurament de la memòria, així com del programari creat i de la presentació.	08/01/2007	08/01/2007
<u>Debat</u> . Període de temps on el tribunal farà les preguntes que cregui pertinents.	15/01/2007	19/01/2007

**Taula 1** Planificació projecte

### 1.5. Productes Obtinguts

Els productes obtinguts en aquest projecte han estat els següents:

- ✓ Memòria. Aquest document mostra els resultats obtinguts tant de la part teòrica com de la part pràctica. També explica com s'ha anat assumint els conceptes necessaris per poder aconseguir els objectius proposats.
- ✓ SIG. S'ha generat un *geoworkspace* amb el programari Geomedia Professional 6.0 on es pot observar la cartografia de Catalunya, així com el model de dades necessari per portar a terme aquest projecte.
- ✓ Comandes. S'han creat les comandes necessàries per resoldre l'enunciat del projecte.

### 1.6. Descripció resta capítols

A continuació s'explica el contingut dels capítols d'aquesta memòria:

- ✓ Capítol 1- Introducció. És aquest capítol i s'explica el motiu del projecte, així com els objectius. També s'exposa quin ha estat el pla de treball, i un breu resum dels capítols d'aquesta memòria.
- ✓ Capítol 2- Introducció al SIG. En aquest capítol s'exposarà quines són les característiques d'un SIG i els seus components. S'exposarà quins tipus de dades s'utilitzen per poder representar el sistema, així com els models de representació i els formats de fitxers utilitzats.
- ✓ Capítol 3- Conceptes cartogràfics i geodèsics. En aquest capítol s'explicaran quins són els conceptes bàsics de cartografia i de geodèsia que es necessiten per poder entendre un SIG i les seves característiques. Per altra banda, s'exposaran els diferents sistemes de projeccions que s'utilitzen en l'actualitat.
- ✓ Capítol 4- Geomedia Professional 6.0. En aquest capítol es farà un breu resum del programari Geomedia Professional 6.0 i s'explicarà quines són les seves característiques i les seves utilitats.
- ✓ Capítol 5- Algoritmes de recerca de camins. En aquest capítol s'explicarà quins són els algoritmes més utilitzats en la recerca de camins i com s'implementen aquests. També s'explicarà quin d'aquests s'ha utilitzat en el projecte i el perquè.
- ✓ Capítol 6- Creació d'un SIG. En aquest capítol s'explicarà "pas a pas" com es construeix un SIG utilitzant Geomedia Professional 6.0.
- ✓ Capítol 7- Implementació de comandes. En aquest capítol s'explicarà tots els passos que s'han seguit per programar les comandes que interaccionen amb Geomedia Professional 6.0. També s'explicarà com s'utilitzen aquestes i quin ha estat el codi font utilitzat.
- ✓ Capítol 8- Conclusions. En aquest capítol s'exposarà quines han estat les conclusions d'aquest projecte i es proposaran noves línies de treball.

Una vegada introduït el projecte, es continuarà amb l'explicació de les característiques d'un Sistema d'informació Geogràfica.

---

---

## 2. INTRODUCCIÓ AL SIG

---

---

En aquest capítol s'explicarà què és un sistema d'informació geogràfica (SIG), i quines són les seves principals característiques (veure apartat 2.1). També s'explicarà quins components ho formen (veure apartat 2.2), així com quines opcions hi ha per poder representar aquest sistema (veure apartat 2.2.3). Per altra banda, també s'explicarà quins formats existeixen avui en dia per emmagatzemar les dades (veure apartat 2.3).

### 2.1. Característiques principals d'un SIG

Un SIG és un sistema informàtic que permet emmagatzemar informació gràfica i no gràfica. Amb aquesta informació es pot organitzar, analitzar i representar objectes situats en la Terra [1].

Un SIG està format per una sèrie de funcions que permeten portar a terme diverses activitats, aquestes es poden definir com [2][4]:

- ✓ Funcions per l'entrada d'informació. Són els processos que faciliten la incorporació d'informació al sistema, tant afegint informació digital com convertint la informació geogràfica i no geogràfica, d'un format analògic a un format digital.
- ✓ Funcions per la sortida d'informació. Són els processos que permeten crear mapes, gràfics i altres tipus d'informació a partir de les dades emmagatzemades en les **bases de dades** (BBDD).
- ✓ Funcions de gestió de la informació. Permet gestionar les dades per poder recopilar només la informació que interessa en cada moment.
- ✓ Funcions analítiques. Permet analitzar les dades de manera que es pot treure informació extra sobre les dades que es tenen emmagatzemades. Aquesta informació pot ser gràfica o no gràfica. Aquestes dades permeten de vegades poder resoldre problemes o prevenir-los.

### 2.2. Components d'un SIG

A part de les funcions (veure apartat 2.1) que permeten interaccionar amb el SIG, existeixen

altres tipus de components que ajuden a realitzar aquestes funcions [2]:

- ✓ Equips. Permeten poder treballar amb les eines adequades pel tractament d'aquest tipus de dades. Es pot treballar amb servidors però també amb ordinadors personals.
- ✓ Programari. Són aplicacions que permeten poder manipular dades, crear mapes. Aquestes aplicacions normalment contenen mòduls que permeten realitzar les funcions abans esmentades.
- ✓ Recursos Humans. Aquesta és una part important, sense aquest component no es podrien actualitzar les dades, ni gestionar-les correctament.
- ✓ Dades. És el conjunt d'informació que descriu els elements que s'utilitzaran més endavant en l'anàlisi del sistema i que s'explicarà més extensament a continuació.
- ✓ Regles de negoci. Tot SIG ha de tenir unes regles de negoci les quals indicaran quins són els objectius del sistema, així com quins passos es seguiran per aconseguir-los.

### 2.2.1. Tipus de dades

A continuació s'explicarà la part més important d'un SIG, les dades. Les dades geogràfiques poden ser adquirides tant per sistemes informàtics, com per persones que ja les tenen emmagatzemades. Les dades emmagatzemen informació sobre els objectes que estan situats a la Terra, aquesta informació té dues parts, per un costat la part espacial i per altra la temàtica de l'objecte [1].

Per tant es pot dir que un objecte geogràfic té dos tipus d'atributs [4]:

- ✓ Geogràfica. Descriu la forma de l'objecte i la seva posició en la superfície terrestre.
- ✓ Temàtica. Serveix per descriure la seva naturalesa i diferenciar-lo de la resta, aquest conjunt d'atributs acostumen a ser alfanumèrics.

### 2.2.2. Conjunt de dades

Una vegada es tenen aquestes dades, els diferents objectes que es tenen s'agrupen depenent

de les seves característiques, aquestes unions s'anomenen categories [1]. Cada categoria agrupa els objectes d'una mateixa temàtica i si es fa una combinació d'aquestes categories es pot arribar a treure un gran anàlisi del sistema.

La següent qüestió que es planteja, és com es representaran aquestes dades. Per un costat, les dades descriptives són fàcils de representar, ja que són dades estàndards, però les dades geogràfiques són més difícils de representar i és per això que es necessitarà d'algun model. Actualment hi han dos tipus de models que fan això i que s'explicaran en el següent punt: el model vectorial i el model *raster*.

### 2.2.3. Models de representació de dades geogràfiques

Existeixen dos tipus de models que permeten representar la informació geogràfica, aquests models són el vectorial i el *raster*. Depenent del tipus d'informació a representar s'escollirà un o un altre. A continuació es definiran les característiques principals de cadascun d'aquests [4]:

#### 2.2.3.1. Model Vectorial

En el model vectorial, els objectes es representen a partir de parells de valors (x,y) que representen la ubicació d'un punt de la Terra. L'agrupació d'aquests punts construeixen les anomenades línies, i la successió de línies construeixen les superfícies. Això fa que aquestes puguin representar figures més complexes. Com tot model té les seves avantatges i les seves desavantatges, com es defineix a la taula 2 [4].

<i>Avantatges</i>	<i>Desavantatges</i>
Es poden obtenir mapes més precisos, ja que es disposa de les coordenades punt a punt.	Fer agrupacions de dades és molt més complicat.
Aquest model agilitza l'anàlisi de xarxes, ja que aquestes estan definides amb punts i línies.	No és adequat per l'anàlisi de superfícies.
Ocupa menys espai físic, ja que només es guarda la informació de l'objecte en concret que es vol representar	No es pot representar límits difusos.
Més senzilla l'actualització de les dades, ja que	

la relació objecte-coordenada és molt més concreta.

**Taula 2** Avantatges i desavantatges del model vectorial

### 2.2.3.2. Model *raster*

En el model *raster* es divideix la superfície a analitzar en una matriu de cel·les o *pixels*, posteriorment s'assigna a cada cel·la un determinat valor que dependrà de la informació temàtica que es vulgui representar i que serà representatiu de tots els elements que englobin les cel·les. S'ha de decidir quin serà la resolució de la matriu, ja que aquesta decisió donarà el nivell de resolució i per tant de qualitat del model. Una matriu amb moltes cel·les donarà una representació més real del terreny, però també serà més difícil de tractar i d'emmagatzemar. Com tot model té les seves avantatges i les seves desavantatges, com es defineix a la taula 3.

<i>Avantatges</i>	<i>Desavantatges</i>
Només es guarda un valor numèric per cel·la, i això simplifica el model.	Requereix un volum important de dades, ja que quan es vol analitzar una superfície, s'ha d'assignar un valor a totes les cel·les.
És un model ideal per analitzar àrees, ja que permet representar fàcilment una determinada superfície.	Poca exactitud ja que la informació que s'obté és molt més difusa.
És més barat representar aquest model.	Inadequat per l'anàlisi de xarxes, ja que les xarxes són punts i línies, i aquest model no guarda de manera precisa aquest conceptes.
Permet representar límits difusos.	

**Taula 3** Avantatges i desavantatges del model *raster*

## 2.3. Formats de fitxers

Tota la informació que es té sobre un SIG s'emmagatzema en fitxers, aquests poden tenir diferents formats depenent de la informació que guardin i del model de representació (veure apartat 2.2.3) . A continuació s'exposaran els diferents formats que es poden trobar.

### 2.3.1. Formats de fitxers pel model vectorial

Els formats més representatius són [5]:

- ✓ DGN: Aquest format està creat per Bentley System, i és utilitzat en aquest projecte.
- ✓ DWG: Format creat per Autocad.
- ✓ DXF: Aquest format està creat per AutoDesk, és un fitxer que intercanvia dades vectorials.
- ✓ EXPORT: Aquest format està creat per ESRI, l'utilitza el SIG Arcinfo creat pel creador abans esmentat.

### **2.3.2. Formats de fitxers pel model raster**

Els formats més representatius són [5]:

- ✓ GeoTIFF: És un format estàndard de metadades que conté una imatge TIFF amb informació georeferenciada.
- ✓ MrSID: Creat per LizardTech. És el software de compressió que genera el format SID, aquest és utilitzat en aquest projecte per descarregar les fotografies que s'insertin en el sistema (veure apartat 6.1.6).
- ✓ TIFF, JPG, GIF: Format estàndards que guarda imatges.

### **2.4. Funcionalitats d'un SIG**

Les aplicacions que pot tenir un SIG són moltes, ja que els camps on es pot aplicar són molt variats. Es poden indicar alguns casos [4][5]:

- ✓ Telecomunicacions. Xarxes de comunicacions, cobertura mòbil, etc.
- ✓ Gas, electricitat, infraestructures urbanes, aigua, canalitzacions, etc.
- ✓ Tràfic. Localització de carreteres, senyals, radars, etc.
- ✓ Topogràfic. Ubicació d'accidents geogràfics, àrees poblades, rius, llacs, tallafocs, etc..



### 3. CONCEPTES CARTOGRÀFICS I GEODÈSICS

---

---

A continuació s'explicarà el concepte de cartografia (veure apartat 3.1) i geodèsia (veure apartat 3.2), aquests conceptes són bàsics per poder entendre com posiciona la informació geogràfica un SIG. Seguidament s'explicarà com un SIG situa la informació i quins mètodes utilitza (veure apartat 3.3).

#### 3.1. Definició de cartografia

La cartografia té com objectiu representar en un plànol la superfície de la Terra (total o parcialment). És per això que estudia els sistemes de projeccions (veure apartat 3.3) que serviran per definir una correspondència matemàtica entre els punts de l'el·lipsoide (veure apartat 3.2.1) i la seva transformació al plànol [10].

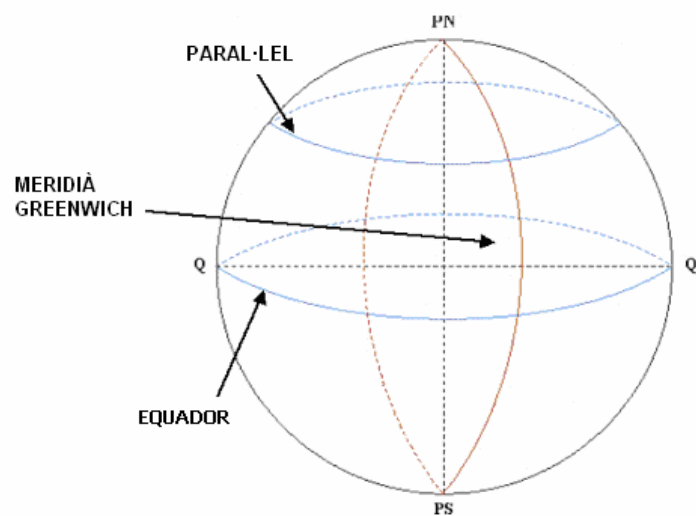
Les coordenades geogràfiques són les dades que representen aquest plànols, les quals s'explicaran a continuació.

##### 3.1.1. Coordenades geogràfiques

Els conceptes bàsics que es defineixen i que permeten situar un punt en un plànol són [5]:

- ✓ Meridians. Són els cercles màxims sobre l'esfera terrestre que passen pels pols, com es pot observar en la figura 1. En coordenades geogràfiques, el meridià de Greenwich és el que s'utilitza com origen de las longituds. Tots el meridians tenen la mateixa longitud. Amb els paral·lels, com s'explicarà més endavant, formen el sistema de coordenades geogràfiques basat en latitud i longitud [7].
- ✓ Paral·lels. És denomina paral·lel al cercle format per la intersecció de l'esfera terrestre amb un pla imaginari perpendicular al eix de rotació de la Terra [8], com es pot observar en la figura 1. Existeixen cinc paral·lels que es corresponen amb una posició concreta de la Terra en la seva òrbita al voltant del Sol y que reben un nom particular:

- Cercle Polar Àrtic (latitud  $66.5^\circ$  N)
- Tròpic de Càncer (latitud  $23.5^\circ$  N). És el paral·lel més al Nord.
- Equador (latitud  $0^\circ$  N)
- Tròpic de Capricorn (latitud  $23.27^\circ$  S). És el paral·lel més al Sud.
- Cercle Polar Antàrtic (latitud  $66.5^\circ$  S)



**Figura 1** Representació meridians i paral·lels [9]

- ✓ Longitud. És l'angle que formen el pla meridià que passa per un punt i el meridià d'origen [5].
- ✓ Latitud. És l'angle que formen el pla que passa per un punt i el centre de la Terra amb el pla de l'equador.

### 3.2. Definició de geodèsia

La geodèsia estudia la forma, les dimensions i el camp gravitatori de la Terra [10]. Estableix una xarxa de punts (vèrtexs geodèsics) distribuïts per la superfície terrestre, que permeten cartografiar grans extensions de terreny.

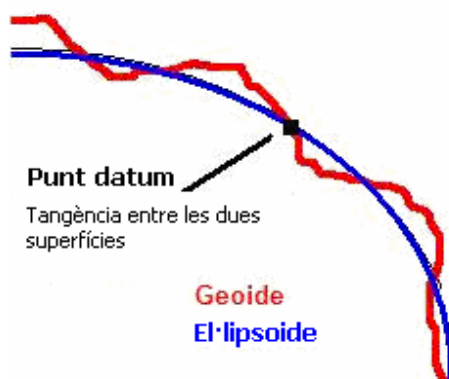
Per poder situar aquest punt s'ha d'escollir un sistema de referència, que en aquest cas serà l'el·lipsoide de revolució, el qual s'explica a continuació.

### 3.2.1. Geoide, el·lipsoide i *datum*

La paraula geoide significa “forma de la Terra”. El geoide és un esferoide tridimensional que constitueix una superfície equipotencial imaginària. El geoide té en compte les anomalies gravimètriques i l'aixafament dels pols, per tant és una superfície irregular (línia vermella en la figura 2). Això, suposa que sigui difícil efectuar càlculs matemàtics sobre aquesta superfície. Per resoldre aquest problema es té que escollir una superfície que sigui matemàticament accessible i és per això que la superfície escollida és un el·lipsoide [10].

Un el·lipsoide (línia blava en la figura 2) és una simplificació del geoide i es defineix de manera matemàtica com un objecte que té com característiques principals dos radis i un aixafament [10], aquest el·lipsoide té en compte els següents factors:

- El centre gravitatori coincideix amb el centre de l'el·lipsoide.
- El pla definit per l'equador terrestre coincideix amb el de l'el·lipsoide.
- La suma dels quadrats de les alçades geoidals (diferència entre geoide i el·lipsoide) sigui mínima.



**Figura 2** Relació entre geoide, el·lipsoide i *datum* [17]

Per poder crear la cartografia d'un lloc, és necessari l'ús d'altre concepte que aporta uns paràmetres de referència que relacionen un punt d'origen del geoide amb l'el·lipsoide. Aquest concepte és el *datum*. El *datum* és aquell punt on l'el·lipsoide i el geoide s'assumeixen com tangents, coincidint així les verticals a les dues superfícies [17], com es pot observar en la figura 2.

Un *datum* és la superfície de referència pel càlcul i determinació de les coordenades geogràfiques (veure apartat 3.1.1). Defineix l'origen i situació d'un sistema de coordenades vàlid per una determinada zona de la Terra [10]. Això pot fer que un mateix punt en la Terra estigui

referenciat de manera diferent segons el *datum* utilitzat. El *datum* que s'utilitza en aquest projecte és l'European 50.

A continuació s'explicarà quins tipus de projeccions es poden fer.

### 3.3. Projeccions cartogràfiques

És impossible reproduir exactament i sense errors un esferoide en un plànol. Ja que la Terra és esfèrica i, per tant, la representació plana de la seva superfície no serà mai real i exacta [13]. Per això es pot dir que una projecció cartogràfica és la tècnica utilitzada per traslladar a un plànol unes dades que estan situades en un el·lipsoide (veure apartat 3.2.1).

Es pot dividir en dos grans grups segons conservin o no les formes i les àrees, això s'anomena distorsions. Així, una projecció és conforme quan conserva la forma i els contorns dels continents, i és equivalent quan representa totes les superfícies en les mateixes proporcions. Si no és conforme ni equivalent se'n diu afillàctica [13].

Hi ha centenars de projeccions diferents, però es poden recollir en quatre grans grups si es té en compte la superfície sobre la qual és projectat el globus. Així es troben les azimuthals o polars, les cilíndriques i les còniques [13]. A continuació s'explicarà més detalladament aquestes.

#### 3.3.1. Projeccions polars

Aquest tipus de projecció s'anomena també azimuthal, o plana. Ja que un pla imaginari és el que toca la superfície del planeta.

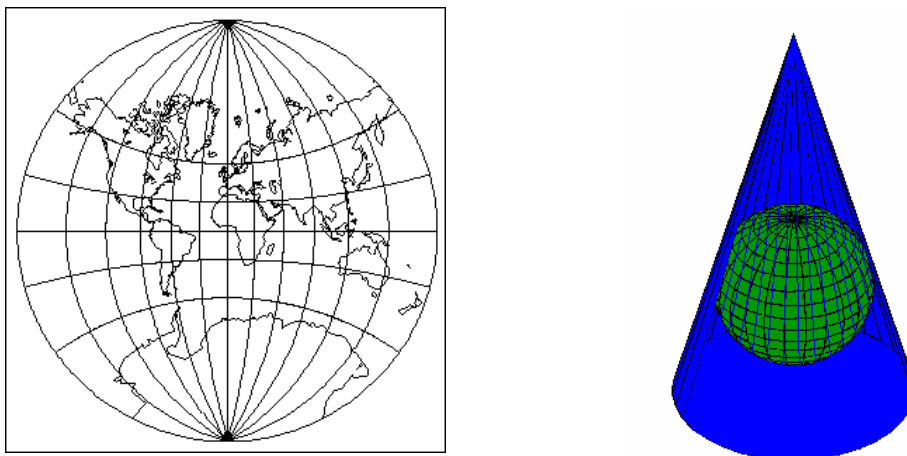
Aquest tipus de projecció es sol fer sobre els pols i donen una imatge com si un observador veiés la Terra des de la vertical del pol. Representen habitualment les zones polars o mostren un hemisferi complet com es pot observar a la figura 3 [13].



**Figura 3** Projectió polar des de l'hemisferi nord [13]

### 3.3.2. Projections còniques

Les projeccions còniques es fan sobre un con que està en contacte amb un paral·lel de latituds mitjanes. La imatge que s'obté és la d'un observador que veu la Terra des de la vertical de les latituds mitjanes, com es pot observar a la figura 4 [13].



**Figura 4** Projectió cònica

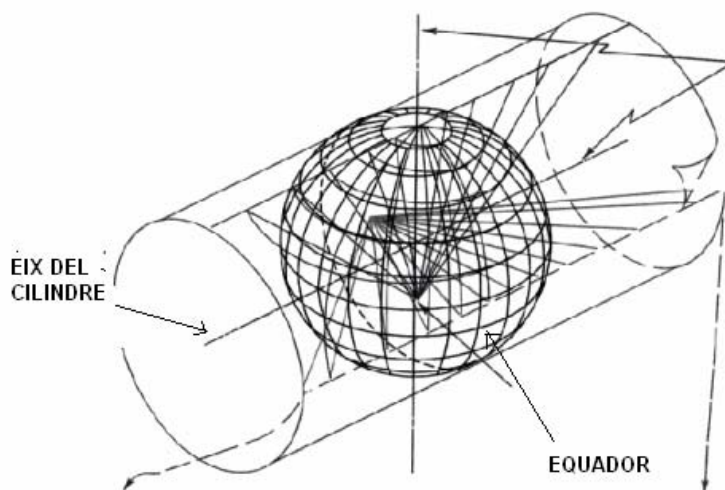
### 3.3.3. Projeccions cilíndriques

Les projeccions cilíndriques es solen fer a l'equador sobre un cilindre que fa la volta a tota la Terra. La imatge que s'obté és la d'un observador que veu la Terra situada en la vertical de l'equador [13].

Aquesta projecció representa bé la zona entre l'equador i els tròpics i després amplia la seva distorsió latitudinalment. La projecció més coneguda és la transversa de Mercator o més coneguda com UTM. A continuació s'explicarà que és la projecció UTM.

### 3.3.4. Projecció UTM

A continuació s'explicarà el sistema de projecció més popular, el sistema *Universal Transverse Mercator* (UTM). És pren com superfície desenvolupadora un cilindre que es col·loca tangent al el·lipsoide de referència, de manera que l'eix del cilindre està dins del pla del l'equador com es pot observar en la figura 5 [10][16].



**Figura 5** Projecció UTM[24]

La projecció UTM no és utilitzada només per les representacions cartogràfiques, sinó també pel sistema de coordenades UTM, les quals seran explicades més endavant, aquest és un sistema de coordenades geogràfiques alternatiu a la utilització de latitud i longitud. Una de les seves

avantatges és que les seves magnituds s'expressen en unitats de longitud [16].

La projecció UTM consta d'un conjunt de coordenades planes, que cobreixen la superfície de la Terra compresa entre els 80° de latitud sud i els 84° de latitud nord. Aquesta superfície es divideix en 60 porcions denominades fusos i 20 porcions denominades zones, els fusos van numerats del 1 al 60 com es pot mostrar en la figura 6. El fus és l'àrea situada entre dos meridians de la Terra, que es distancien entre ells 6° longitudinals [20].

La projecció UTM té l'avantatge de que cap punt està allunyat del meridià central de la seva zona, per tant les distorsions són petites. Però això s'aconsegueix mitjançant la discontinuïtat: un punt en el límit de la zona es projecta en dos punts distints, a no ser que estigui en l'equador. Una línia que uneixi dos punts entre zones contigües no és contínua a no ser que crui per l'equador.

Per evitar aquestes discontinuïtats, de vegades s'estenen les zones, per que el meridià tangent sigui el mateix. Això permet plànols continus que són molt compatibles amb els estàndards . Encara que, els límits d'aquestes zones, les distorsions són majors que en les zones estàndards.

#### **3.3.4.1. Coordenades UTM**

El sistema de coordenades utilitzat en la projecció UTM, rep el nom de coordenades UTM, i sempre estan expressades en metres. A l'hora de tractar amb coordenades UTM s'ha de tenir en compte el fus (veure apartat 3.3.4) en el que es troba la zona del plànol. Les coordenades UTM tenen un sistema de referència completament diferent en cada fus, amb això s'ha aconseguit disminuir les distorsions produïdes per aquest tipus de projecció [20].

Espanya es troba principalment al fus 30 (de 6° oest a 0°), però també té zones en el fus 31 i el 29, com és el cas de Catalunya que es troba en el fus 31. Respecte a les zones, Espanya es troba entre les zones T i S, en concret Catalunya es troba a la T.

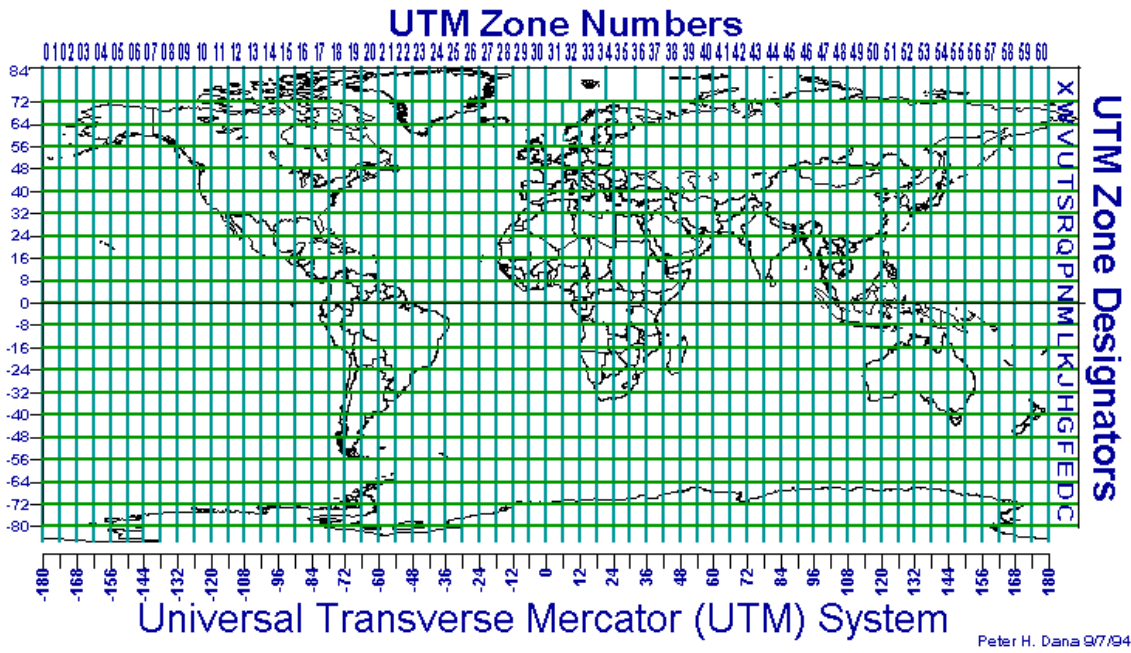


Figura 6 Coordenades UTM



---

---

## 4. GEOMEDIA PROFESSIONAL 6.0

---

---

En aquest capítol s'exposarà una breu introducció al funcionament de Geomedia Professional 6.0 d'Intergraph. Per poder aprendre a utilitzar aquest programari són necessaris els coneixements adquirits en els capítols anteriors, tal com quins tipus de projeccions i coordenades es poden utilitzar (veure apartat 3.3).

A continuació s'explicarà quines són les característiques generals d'aquest programari, així com quins són els seus elements principals i el seu funcionament.

També es veurà com aquest programari pot interaccionar amb llenguatges de programació estàndards per poder crear noves funcionalitats. En el cas d'aquest projecte amb Visual Basic 6.0 de Microsoft, però també amb altres llenguatges com Delphi.

### 4.1. Introducció a Geomedia Professional 6.0

Geomedia Professional 6.0 és un SIG (veure apartat 2), on els seus components es representen en Geomedia Professional 6.0 de la següent manera [18]:

- ✓ Dades. Geomedia Professional 6.0 permet connectar a varis sistemes de BBDD com ara Microsoft Access, Oracle, Microsoft SQL Server. També pot importar dades altres sistemes com ara el CAD, com s'explicarà en l'apartat 6.1.5.
- ✓ Programari. Geomedia Professional 6.0 disposa d'eines gràfiques per mostrar la informació geogràfica, per visualitzar plànols, etc..

Aquest programari també permet poder utilitzar algunes de les funcions que té un SIG, com ara:

- ✓ Funcions de gestió de la informació. Té eines per poder vectoritzar i digitalitzar imatges.
- ✓ Funcions analítiques. Permet crear consultes sobre les dades georeferenciades, etc...

#### 4.2. Elements principals de Geomedia Professional 6.0

Geomedia Professional 6.0 té uns elements principals que fan possible la utilització i l'aprofitament d'aquest. En aquest cas els elements principals d'aquest programari són els següents [5][18]:

- ✓ Geoworkspace. És l'element en el qual es porta a terme el treball que s'està realitzant, es pot dir que és la base del programari, ja que emmagatzema totes les connexions a les dades externes, així com la definició de la llegenda i les personalitzacions de cada SIG en particular.
- ✓ Magatzem. En aquest element es troben les connexions a les dades externes.
- ✓ Classe d'entitat. Defineix tots els atributs i tipus de dades associades a les entitats, de manera que resulta més fàcil la gestió d'entitats.
- ✓ Entitat. Les entitats són representacions gràfiques d'elements del món real. Es poden representar com punts, línies, àrees, texts i imatges.
- ✓ Llegenda. És l'element que controla tota la informació que apareix en la finestra de treball.
- ✓ Consultes. Les consultes es realitzen per treure informació del sistema que s'està construint.
- ✓ Sistemes de coordenades. Aquests sistemes permeten relacionar les entitats descrites amb punts en l'espai real.

A continuació s'explicarà com aquests elements s'utilitzen en el programari.

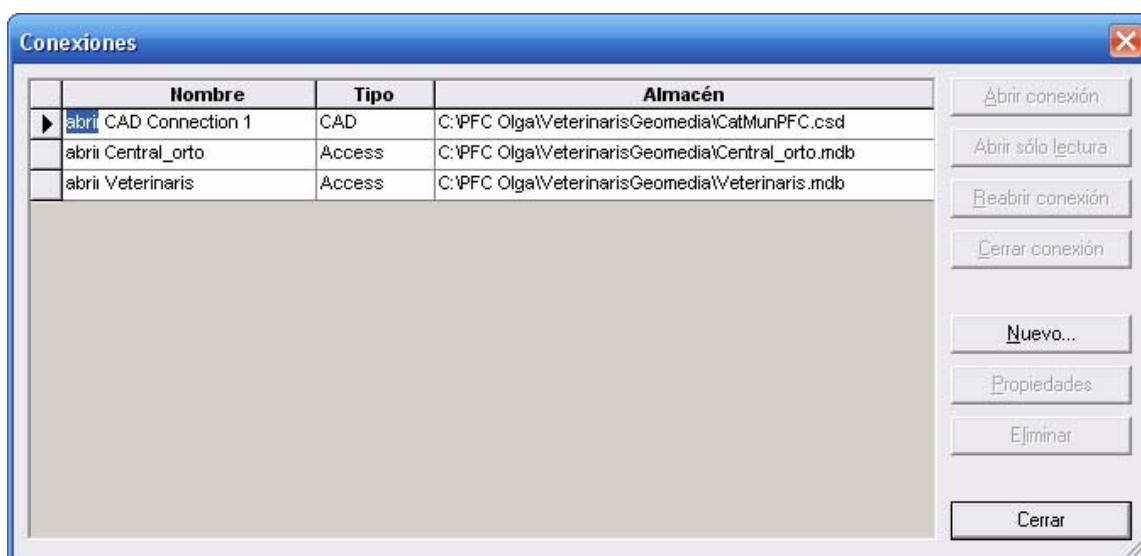
#### 4.3. Utilització de Geomedia Professional 6.0

A continuació s'exposarà els diferents elements del Geomedia Professional 6.0 i com és el seu funcionament.

##### 4.3.1. Connexió a les dades

En Geomedia Professional 6.0 es necessita de dades per poder mostrar informació en els

plànols, així com per poder realitzar un anàlisi o una consulta, com s'ha vist al apartat 4.1. Aquestes dades es guarden en els magatzems. Aquest fitxers són externs a Geomedia Professional 6.0 i es connecten mitjançant una connexió a unes dades externes, com es pot veure en la figura 7, en aquest cas Microsoft Access i CAD, però també podria ser Microsoft Sql Server, Oracle, i d'altres [18].



**Figura 7** Llistat de connexions actives a Geomedia Professional 6.0

#### 4.3.2. Sistema de coordenades

Els sistemes de coordenades amb els que treballa Geomedia Professional 6.0 són [5]:

- ✓ Geogràfic. Expressa les coordenades en forma de longitud y latitud.
- ✓ Projectat. Expressa les coordenades en forma de x i y (est i nord respectivament). Es refereix a un pla de projecció amb una relació amb l'esferoide coneguda.
- ✓ Geocèntric. Es refereix a un sistema de coordenades cartesianes amb el centre de la Terra, que expressa les coordenades(x,y,z) com la posició d'un punt específic respecte a aquest punt.

Aquests sistemes poden variar entre els de magatzem i el del *geoworkspace*, situació que fa que sigui molt important revisar aquestes dades per tal de no crear problemes en la visualització.

### 4.3.3. Mapes i dades

La finestra de dades permet visualitzar les entitats que es tenen agregades a la llegenda (veure apartat 4.2), aquestes entitats les visualitza mitjançant un plànol, com es pot veure a la figura 8.

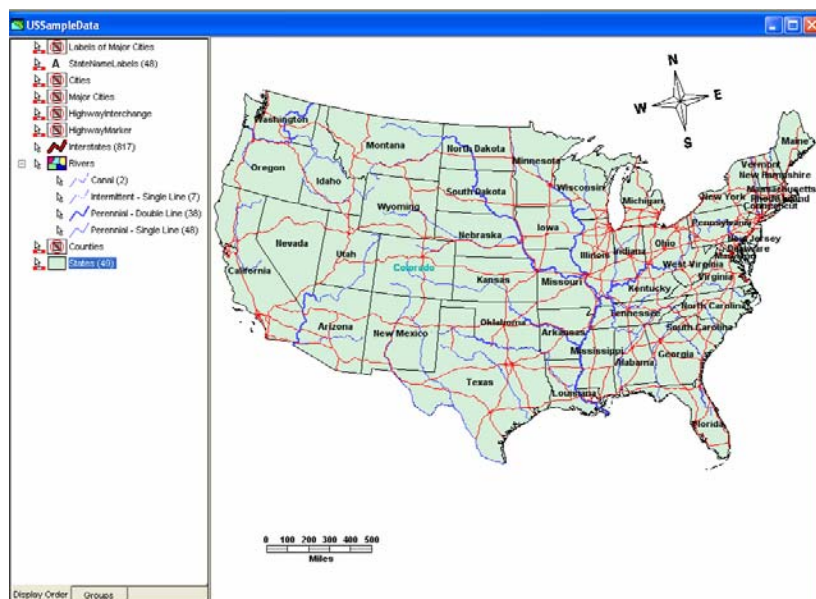


Figura 8 Finestra de visualització de dades

### 4.3.4. Entitats

Les entitats que Geomedia Professional 6.0 permet crear són les següents [5]:

- ✓ De punt. Es representen per mitja d'un punt en el plànol, també poden representar entitats que no es poden visualitzar de millor manera.
- ✓ De Línia. Es representa mitjançant un o més arcs o línies, com per exemple els rius de la figura 8.
- ✓ D'àrea. Es representa mitjançant zones delimitades, com per exemple els estats de la figura 8.
- ✓ D'imatge. Són imatges raster.
- ✓ Compostes. Unió de varius tipus d'entitats en una.

- ✓ De Text. Són etiquetes posicionades en algun lloc del plànol, com per exemple els noms dels estats a la figura 8.

Aquestes entitats poden ser introduïdes manualment o importades des d'algun sistema com es veu en l'apartat 4.3.5

#### **4.3.5. Captura i correcció de dades**

Geomedia Professional 6.0 proporciona eines per efectuar la captura de les dades (veure apartat 6.1.5), de manera que poden agafar diferents formats de fitxers (veure apartat 2.3) i aconseguir visualitzar-los en el programari [5].

Aquesta captura de dades de vegades pot ocasionar algun error de connectivitat i geometria, però aquest programari inclou eines per poder solucionar aquests errors.

#### **4.3.6. Anàlisis**

Amb Geomedia Professional 6.0 es poden efectuar consultes, establir relacions i fer anàlisis de la geometria. Les consultes es poden realitzar de dues maneres: per filtres d'atributs o per consultes espacials, i també ambdues.

Les primeres retornen entitats que compleixen uns requisits establerts. Les segones retornen entitats que mantenen una relació espacial amb altres. Un exemple de la primera seria saber quants estats d'EEUU tenen més de 5000 habitants, mentre que el segon tipus de consulta diria quins estats estan al costat o es toquen.

Totes aquestes consultes i funcions es poden automatitzar via programació a partir de Visual Basic 6.0, tal i com s'explicarà a continuació.

#### **4.4. Interacció amb Visual Basic 6.0**

Geomedia Professional 6.0 és un programari que permet automatitzar funcions des de llenguatges de programació estàndards. Des d'aquests llenguatges es podran crear aplicacions dins del propi programari, ja que integra completament les aplicacions creades en el seu entorn de

treball.

També es podrà accedir a totes les funcionalitats de Geomedia Professional 6.0, així com utilitzar els controls estàndards dels llenguatges de programació per tal de fer una aplicació molt més àgil i amena per l'usuari. S'ha escollit Microsoft Visual Basic 6.0 per l'experiència aportada amb aquesta eina.

Els passos a seguir per poder crear comandes són els següents:

- ✓ Pas 1: El llenguatge de programació és Microsoft Visual Basic 6.0. amb el *Command Wizard* de Geomedia Professional 6.0.
- ✓ Pas 2: Una vegada creada la comanda es té que generar una DLL, aquesta DLL serà la que permetrà interaccionar amb Geomedia Professional 6.0.
- ✓ Pas 3: Per tal de que Geomedia Professional 6.0 reconegui aquesta DLL es té que registrar. Per registrar aquesta DLL s'utilitzarà la següent instrucció:

```
InstallAppCmd /cmdset Planificar /loc C:\PFC Olga\VeterinarisGeomedia\PlanificarRuta\bin /dll  
C:\PFC Olga\VeterinarisGeomedia\PlanificarRuta\bin
```

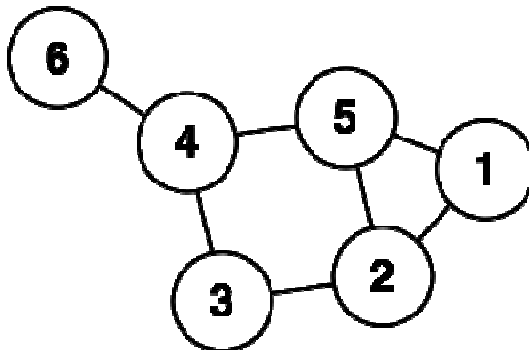
- ✓ Pas 4: Una vegada registrada es pot utilitzar des de Geomedia Professional 6.0 afegint aquesta comanda a la barra d'eines del programari.

## 5. ALGORITMES DE RECERCA DE CAMINS

En aquest capítol es farà una petita introducció als algoritmes de recerca de camins. En l'apartat 5.1 s'explicarà una breu introducció a la teoria dels grafs i com aquesta té relació amb el projecte. En l'apartat 5.2 s'explicarà quins tipus d'algoritmes existeixen actualment per resoldre el problema de trobar un camí entre dos punts d'un graf.

### 5.1. Teoria de Grafs

El problema de cercar un camí entre dos punts es pot relacionar fàcilment amb la idea dels grafs. Un graf és un conjunt de nodes relacionats entre ells per línies imaginàries. En el cas d'aquest projecte els nodes serien les granges, i les connexions serien els camins utilitzats per arribar a elles des del node origen. Es pot observar en la figura 9 com és la representació d'un graf.



**Figura 9** Representació d'un graf

En la teoria de grafs aquestes connexions s'anomenen arestes, i poden estar o no ponderades. La diferència entre estar ponderades o no, és que una aresta sense ponderar no dóna informació extra sobre la connexió, en canvi una aresta ponderada dóna informació extra, com ara el temps que es triga en recorre la distància d'un node a un altre [15].

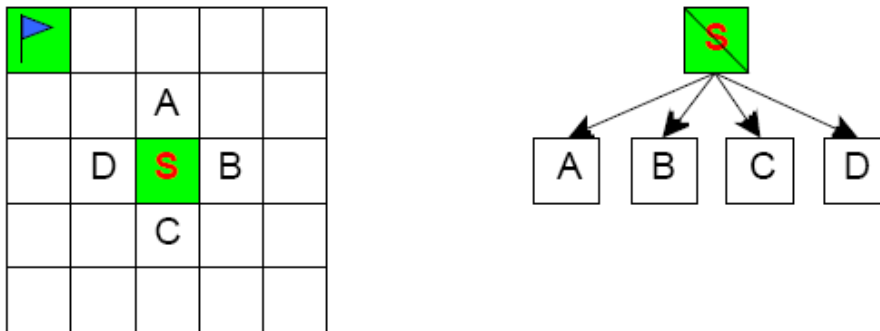
En aquest projecte es vol aconseguir el poder obtenir un camí per fer el recorregut entre dos punts, per això caldrà plantejar algoritmes de recerca de camins. Aquests algoritmes s'expliquen a continuació.

### 5.2. Tipus d'algoritmes de recerca de camins

Tots els algoritmes de recerca de camins tenen un objectiu comú que és, donar la solució de com arribar des d'un punt d'origen a un punt destí. Per tal de resoldre aquest problema s'han plantejat varis mètodes per recorre un graf. A continuació s'explicarà alguns d'aquests mètodes.

#### 5.2.1. Algoritme Profunditat - Primer

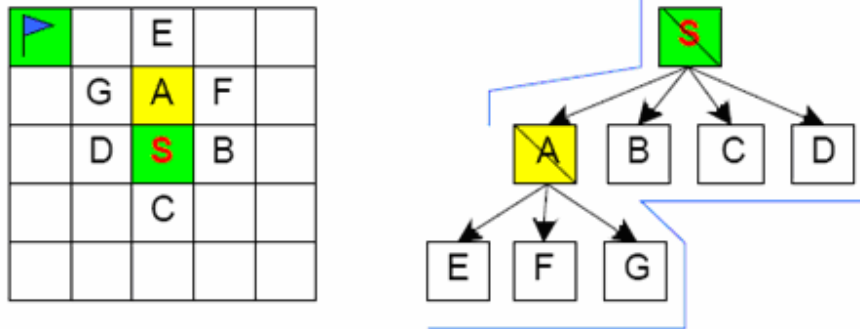
Aquest algoritme és no admissible, això vol dir que podria existir una solució i mai trobar-la. De tots el moviments possibles que es poden estipular per resoldre el problema aquest algoritme sempre escollirà continuar pel node que estigui a més profunditat [26].



**Figura 10** Estat inicial de l'algoritme profunditat - primer i els seus possibles moviments

Com es pot observar el la figura 10, primer de tot s'identifiquen tots el possibles moviments i es comprova que no siguin cap d'ells el destí. En el cas de no ser el destí, aquests moviments es marcaran com possibles. Una vegada marcats s'escollirà el que estigui a més baix nivell, en el cas de que tots estiguin al mateix nivell s'escollirà segons una norma que es crearà internament. Una vegada escollit el moviment es tornarà a processar quins moviments són possibles des d'aquest nou node (com es pot observar a la figura 11) i es marcaran com possibles, a no ser que siguin el node destí, a la mateixa vegada es marcarà com visitat el node on s'està situat. Això es continuarà fins arribar a la solució, si és possible.





**Figura 11** Segon estat i els seus possibles moviments

Es descriu a continuació el pseudocodi que generaria aquest tipus d'algoritme:

```

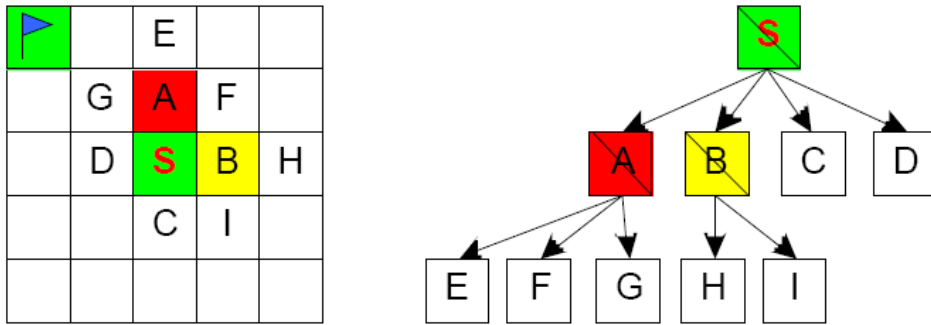
ListaNodes LNodesPossibles
ListaNodes LNodesMoviments
ListaNodes LNodesMarcats
Grafo Gr
Node NodeActual
NodeActual=Origen(Gr)
LNodesMoviments(Gr,NodeActual)
Mentre NoDesti(LNodesMoviments)
    InserirNodes(LNodesPossibles, LNodesMoviments)
    InserirNode(LNodesMarcats,NodeActual)
    NodeActual=EscollirMesProfun(LNodesPossibles)
    LNodesMoviments(Gr,NodeActual)
FiMentre
Mostrar(LNodesMarcats)
    
```

**5.2.2. Algoritme Amplada - Primer**

Aquest algoritme és admissible, això vol dir que en el cas d'existir una solució, la trobarà, encara que no sigui la més òptima, que vol dir que és possible que la solució trobada no sigui la més convenient [26].

A diferència de l'algoritme explicat anteriorment, el qual escollia el node més profund, aquest escollirà sempre el node que estigui a més alt nivell dins del graf. El inici de l'algoritme seria el

mateix que abans (veure figura 10 i 11), La diferencia està a l'hora d'escollir per segona vegada, ja que aniria al node que està a més alt nivell com es pot observar en la figura 12.



**Figura 12** Selecció de nodes de l'algoritme d'amplada - primer

Es descriu a continuació el pseudocodi que generaria aquest tipus d'algoritme:

```

ListaNodes LNodesPossibles
ListaNodes LNodesMoviments
ListaNodes LNodesMarcats
Grafo Gr
Node NodeActual
NodeActual=Origen(Gr)
LNodesMoviments(Gr,NodeActual)
Mentre NoDesti(LNodesMoviments)
    InserirNodes(LNodesPossibles, LNodesMoviments)
    InserirNode(LNodesMarcats,NodeActual)
    NodeActual=EscollirMesAltNivell(LNodesPossibles)
    LNodesMoviments(Gr,NodeActual)
FiMentre
Mostrar(LNodesMarcats)
    
```

### 5.2.3. Algoritme A\*

Aquest algoritme és un dels algoritmes més utilitzat, ja que per una banda és admissible i a més és òptim. A diferencia dels altres algoritmes explicats, aquest basa els seus moviments en l'avaluació d'una funció del tipus [25]:

$$f(n) = g(n) + h(n)$$

On:

- ✓  $g(n)$  : és el cost dels moviments realitzats.
- ✓  $h(n)$  : és la funció heurística. Representa el cost d'escollir un node o un altre.

En el següent exemple es representa com funciona aquest algoritme. Es té un estat inicial com mostra la figura 13, on es tenen obstacles marcats en negre. Quan es comença l'algoritme al igual que en els altres, s'escull tots els possibles moviments i és a l'hora d'escollir quin moviment es farà quan la funció heurística determina quina és la millor opció, que sempre serà la de menor valor, com mostra la figura 14. Així successivament fins arribar el destí.

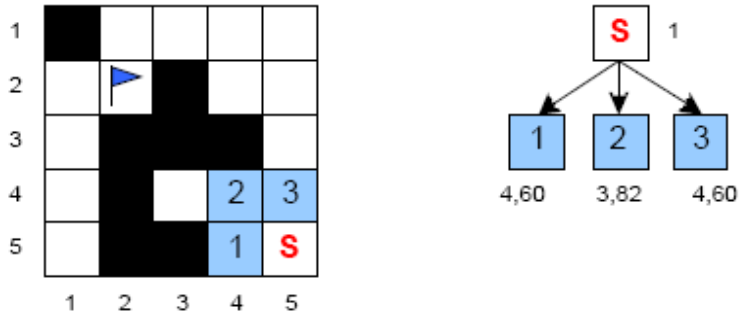


Figura 13 Estat inicial algoritme A\*

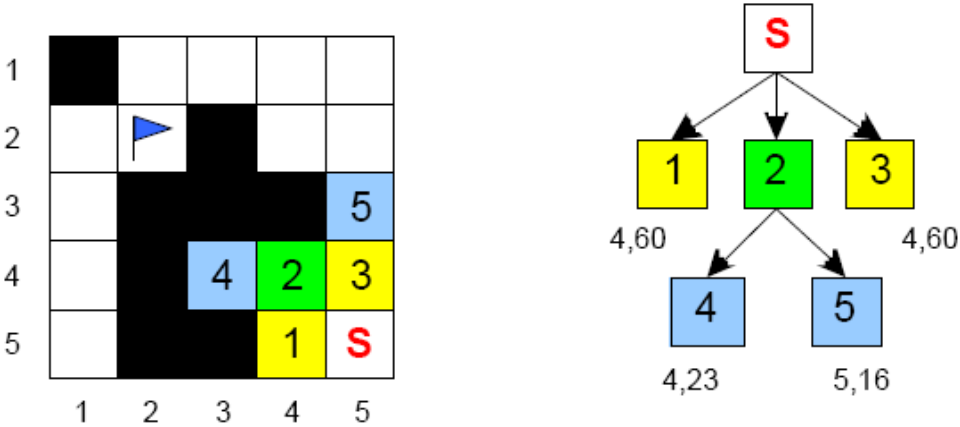


Figura 14 Valors de la funció heurística dins del graf

Es descriu a continuació el pseudocodi que generaria aquest tipus d'algorisme:

```

        ListaNodes LNodesMoviments
        ListaNodes LNodesVisitats
        Grafo Gr
        Node NodeActual
        NodeActual=Origen(Gr)
        LNodesMoviments(Gr,NodeActual)
        Mentre NoDesti(LNodesMoviments)
            InserirNode(LNodesVisitats,NodeActual)
            NodeActual=MenorValor(LNodesMoviments)
            LNodesMoviments(Gr,NodeActual)
        FiMentre
        Mostrar(LNodesVisitats)
    
```

### 5.3. Algorisme escollit

L'algorisme escollit ha estat "Amplada - Primer". El motiu pel qual s'ha escollit aquest és per una banda que és admissible (veure apartat 5.2.2), per tant trobarà una solució, encara que no sigui la més òptima. I per altra banda és la simplicitat de l'algorisme mateix, ja que no ha de calcular cap tipus de funció matemàtica que retardi el procés.

Per últim es mostra l'algorisme implementat, el codi font i l'explicació d'aquest es pot veure a l'apartat 7.2.2.6.

---

```

' El primer node serà l'Origen
ListaNodosPosibles=InsertarInicio_NodosPosibles(Ox, Oy, Oz)
'Mentre troba el destí en continua amb la recerca
While Not EncontradoDestino
'Agafen al següent node la de llista de nodes possibles, si el primer cop s'agafa l'origen
    ArrayNodos = IrASiguiente(ListaNodosPosibles)
    Nx = ArrayNodos(0)
    Ny = ArrayNodos(1)
    Nz = ArrayNodos(2)
    PadreNodo= ArrayNodos(3)
    OrdenBusqueda= ArrayNodos(4)
'es busca les rutes on aquest node existeix
    Rutas=BuscarRutas(Nx, Ny, Nz)
's'hi hi ha rutes continuem
    If ExistenRutasNodos Then
    
```

---

---

```

'es busquen els nous nodes possibles en cadascuna de les rutes que s'han trobat
Nodos=BuscarSiguientsNodos(Rutas)
I=0
'Si es troba que es pot anar a més nodes es continua
If ExistenNodos(Nodos) Then
  'Es recorre tots el nodes nous possibles
  While (ExistaNodos(Nodos(i)))
    'Si el node no és el destí
    If Not DestinoOK(Nodos(i))
      ' i tampoc existeix com a node possible, en el cas de que '
      ' existeixi no l'afegim
      If Noexiste(Nodos(i),ListaNodosPosibles) Then
        AnyadirNodosPosibles(Nodos(i))
      End if
    Else
      'Si és el destí es finalitza i s'afegeix al nodes
      'possibles el destí trobat
      EncontradoDestino = True
      AnyadirNodosPosibles(Nodos(i))
    End If
    I=i+1
  Wend
End If
End If
Wend
'Una vegada trobat,es recorre la llista enredera per trobar el camí.
CrearRuta(ListaNodos)

```

---

---

---

## 6. CREACIÓ D'UN SIG

---

---

En aquest capítol s'explicarà que és l'Institut Cartogràfic de Catalunya (ICC) i per què serveix (veure apartat 6.1), també s'explicarà com es crearan cadascuna de les parts del SIG (veure apartat 6.2, 6.3 i 6.4), mostrant els passos més importants i les decisions preses en cada moment.

### 6.1. Disseny i creació d'un SIG en l'àmbit de Catalunya

Per poder crear un SIG de Catalunya a Geomedia Professional 6.0 caldrà obtenir informació cartogràfica de Catalunya, aquesta informació es pot aconseguir en l'ICC (veure apartat 6.1.1), també s'explicarà quines passes s'han seguit per construir el SIG a l'àmbit de Catalunya a partir de les dades descarregades.

#### 6.1.1. L'ICC

L'ICC és una entitat pública de la Generalitat de Catalunya adscrita al Departament de Política Territorial i Obres Públiques. Va ser inaugurat al 1982, i la seva finalitat es portar a terme estudis sobre cartografia i geologia a Catalunya. És també el responsable de disposar de la informació necessària per dur a terme accions territorials [5].

#### 6.1.2. Finalitats de l'ICC

L'ICC té com a finalitat dur a terme les tasques tècniques de desenvolupament de la informació cartogràfica i geològica, com per exemple [21]:

- ✓ Elaboració, reproducció i difusió de treballs cartogràfics de base, que es concreta en programes d'actuació sobre tot a Catalunya.
- ✓ Densificació i conservació de la xarxa geodèsica d'ordre inferior.
- ✓ Elaboració dels projectes de cartografia vial necessaris per a la realització dels projectes de carreteres i obres públiques a Catalunya.

- ✓ Execució de programes de desenvolupament de la cartografia temàtica i la destinada a l'avaluació de recursos, mitjançant tècniques de teledetecció per a l'estimació d'àrees afectades per incendis, usos del sòl, geologia, etc.
- ✓ Creació, estructuració i organització de la Cartoteca de Catalunya, la qual coordina la recollida i l'estudi de la documentació geogràfica i cartogràfica existent.
- ✓ Formació d'un banc de dades cartogràfiques amb la finalitat d'utilitzar sistemes automàtics en el traçat de la cartografia, que permet no solament l'obtenció de cartografia de base, sinó també l'explotació immediata per a serveis com les obres públiques, el cadastre, etc.
- ✓ Coordinació tècnica dels treballs cartogràfics que facin entitats públiques i privades, si s'escau, i la col·laboració amb organismes públics, tant de comunitats autònomes com a nivell d'estat, i entitats privades d'anàloga finalitat.
- ✓ Publicació i difusió dels treballs que es creguin d'interès públic o científic realitzats pel institut Cartogràfic.

### 6.1.3. Catàleg de l'ICC

L'ICC té un catàleg a disposició pública on es pot trobar dades cartogràfiques i geològiques de Catalunya. Aquestes dades estan emmagatzemades en fitxers (veure apartat 2.3.1) del següent format [5]:

- ✓ DGN i DXF per al model vectorial. Aquest format s'utilitza per descarregar tota la informació geogràfica del territori de Catalunya.
- ✓ MrSID pel model raster. Aquest format s'utilitza per descarregar la ortofoto de la central del laboratori.
- ✓ ASCII per les elevacions de terreny. En aquest projecte no s'utilitza aquest format.

L'ICC és utilitzat en aquest projecte per descarregar tota la informació necessària.

#### 6.1.4. Crear un *geoworkspace*

Una vegada es té la informació sobre la geografia catalana, cal començar a treballar en la incorporació d'aquestes dades al sistema que s'està creant. Primer de tot cal crear un *geoworkspace* (veure apartat 4.2) en el qual s'ha emmagatzemaran totes les dades. Per tal de poder importar les dades aconseguides en el ICC (veure apartat 6.1.1) s'ha d'utilitzar les utilitats que Geomedia Professional 6.0 ofereix, com es veurà més endavant. El resum del *geoworkspace* es mostra en la següent taula:

Propietat	Valor
Nom del fitxer <i>geoworkspace</i>	Veterinaris.gws
Plantilla	UTM31.gwt
Unitat de mesura	1 m
Tipus de projecció	Universal Transversa Mercator (UTM)

**Taula 4** Taula amb las característiques del *geoworkspace*

#### 6.1.5. Importació de les dades de l'ICC

Geomedia Professional 6.0 permet importar dades creades en altres sistemes, tal i com es va explicar en l'apartat 4.3.5. A continuació s'explicarà com es fa pas per pas, de manera que quedi clar com s'ha fet la importació de les dades [5][22][23]:

- ✓ **Creació arxiu coordenades:** Primer de tot cal crear un arxiu amb les coordenades que s'utilitzarà, per fer això s'agafarà el fitxer de CAD de l'ICC i se li canviarà l'extensió per *.csf*, això permetrà que automàticament ja es tingui un fitxer amb el sistema de coordenades correcte. Una vegada es té el fitxer, s'obre aquest amb la utilitat "Definir archivo de sistema de coordenadas" com es pot observar en la figura 15. En aquesta utilitat es defineixen els següents paràmetres com es mostra en la taula 5:

Paràmetre	Valor
Algoritme de projecció	UTM Zona 31 (Catalunya), veure figura 16.
Nom del fitxer	Catalunya.csf
Unitat de mesura	1 m

**Taula 5** Taula amb el paràmetres de l'arxiu de coordenades



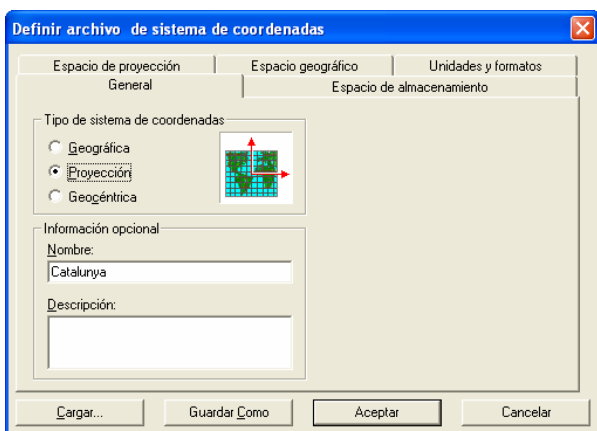


Figura 15 Pantalla de la utilitat “Definir archivo de sistema de coordenadas”

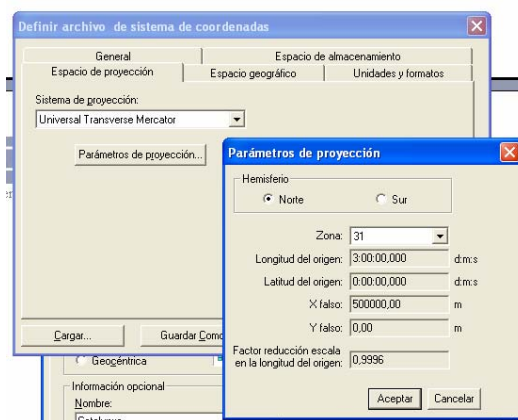


Figura 16 Pantalla de canvi de l'algoritme de projecció

Una vegada ja es té el fitxer de coordenades es crearà l'arxiu de CAD, com s'explica a continuació.

- ✓ **Creació arxiu amb esquema CAD:** Una vegada es té el fitxer de coordenades creat, cal crear un arxiu on s'hi afegirà l'esquema CAD, això es fa des de la utilitat de Geomedia Professional 6.0 que es diu “Definir archivo de esquema de servidores CAD”. Aquesta utilitat primer demana quins fitxers .dgn (veure apartat 2.3.1) es volen agregar a aquest esquema i després quines coordenades s'utilitzaran, en aquest cas s'especificarà el fitxer abans creat (veure taula 5). El fitxer resultant es diu “CatMunPFC.csd”. Una vegada definit quins fitxers s'hi importaran, cal crear un grup d'entitats (veure apartat 4.3.4) per tal de poder mostrar la informació, aquestes entitats s'expliquen a continuació.
- ✓ **Definir Entitats:** El següent pas és definir les entitats que s'utilitzaran en el sistema, aquesta informació es trobarà en els fitxers que l'ICC (veure apartat 6.1) proporciona, les entitats que es creen tenen les següents propietats (com es pot observar en la taula 6) :

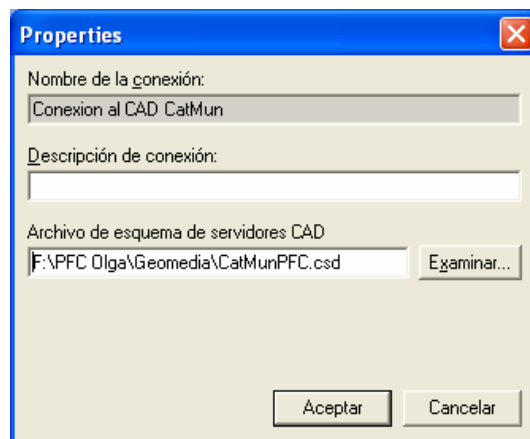
Nom Entitat	Tipus de Geometria	Criteri
Enclavament	De Punt	CellName='ENCL'
Illa	De Punt	CellName='ILLA'
LimitMunCA	De Línia	ElementLevel=3
LimitMunCom	De Línia	ElementLevel=2

LimitMunEstados	De Línia	ElementLevel=4
LimitMunGen	De Línia	ElementLevel=1
LimitMunCosta	De Línia	ElementLevel=5
LocCapComarca	De Punt	CellName='CAPC'
LocCapMunicipi	De Punt	CellName='CAPM'
NomComarca	De Text	ElementLevel=28
NomMunicipiCapCom	De Text	ElementLevel=21
NomMunicipiGen	De Text	ElementLevel=20
NomMunicipiNoCap	De Text	ElementLevel=25

**Taula 6** Taula de propietats de les entitats de la base municipal de Catalunya

El següent pas és crear des de Geomedia Professional 6.0 una connexió a aquest nou fitxer.

- ✓ **Crear connexió CAD:** El següent pas és crear una connexió des de Geomedia Professional 6.0 amb aquest fitxer, per això es crearà una nova connexió amb les dades com es pot veure a la figura 17, de manera que s'heretarà les entitats definides en el *geoworkspace*. Una vegada fet això caldrà modificar les escales de visualització de les entitats, per tal que només es vegin en la representació segons l'escala en que s'està treballant, com fer això s'explicarà en el següent punt.



**Figura 17** Dades de la connexió a un fitxer CAD

- ✓ **Definir escales de visualització:** A continuació es mostra les escales que s'han definit (veure taula 7), es pot observar en la figura 18 i 19 quina diferencia hi ha de visualització entre una escala i un altra.

Nom Entitat	Escala mínima	Escala Màxima
Enclavament	1	25.000.000
Illa	1	25.000.000
LimitMunCA	1	1.000.000.000
LimitMunCom	1	1.000.000.000
LimitMunEstados	1	1.000.000.000
LimitMunGen	1	20.000.000
LimitMunCosta	1	1.000.000.000
LocCapComarca	1	25.000.000
LocCapMunicipi	1	25.000.000
NomComarca	1	200.000.000
NomMunicipiCapCom	1	25.000.000
NomMunicipiGen	1	25.000.000
NomMunicipiNoCap	1	25.000.000

Taula 7 Descripció de les escales utilitzades per cada entitat

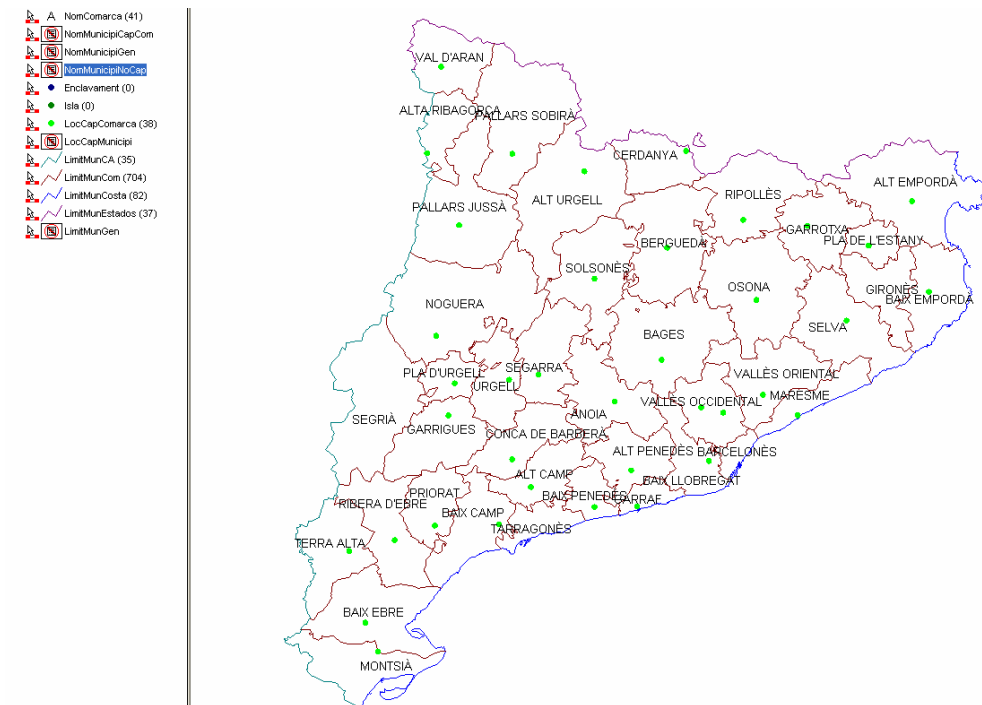


Figura 18 Visualització a escala 1:111.853.762

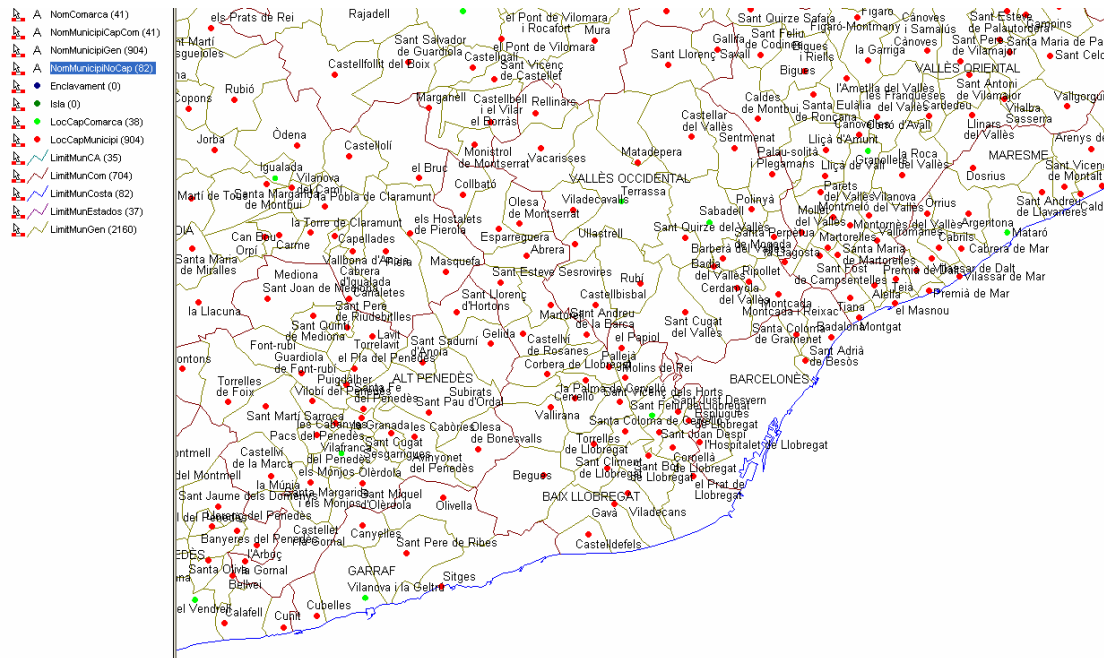


Figura 19 Visualització a escala 1:13.981.720

Una vegada es tenen les dades cartogràfiques importades, s'explicarà com es pot importar fotografies al SIG.

### 6.1.6. Ortofotos de la zona de Lleida

Una ortofoto és una fotografia aèria vertical que ha estat rectificadament de tal manera que es manté una escala uniforme a tota la superfície de la imatge i constitueix una representació geomètrica a escala de la superfície terrestre. Aquestes fotografies es poden aconseguir gratuïtament en l'ICC [21].

Per tal de complementar el sistema s'ha escollit una fotografia de la central del laboratori, aquesta estarà ubicada en Juncosa (veure apartat 7.1).

Les ortofotos són fotografies que reproduïxen part del territori, el territori de Catalunya està compost per tot un grup d'ortofotos. Cada fotografia que es descarrega ve identificada bàsicament per la seva escala i les dues coordenades del vèrtex superior esquerra. Cada ortofoto es compon de dos fitxers: un fitxer amb extensió *.sid* i altre amb extensió *.sdw*. Aquest últim conté la precisió i les coordenades. El fitxer *.sid* és la fotografia [5]. Encara que també es pot descarregar en format TIFF, encara que en aquest projecte no és el cas.

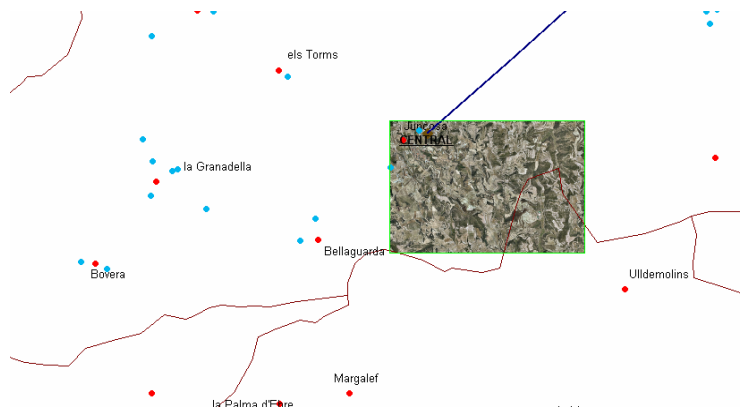
Per incloure ortofotos en el sistema s'han de seguir els següents passos:

- ✓ Descarregar la imatge de l'ICC.
- ✓ Identificar les entrades pel càlcul de coordenades, aquestes dades es trobaran en els dos fitxers abans comentats.
- ✓ Calcular les coordenades dels 3 punts restants, com es pot veure a la taula 8

Alçada	5005,00
Amplada	7288,00
Precisió	0,50
x1	313573,75
y1	4583043,25
x2	320861,75
y2	4583043,25
x3	320861,75
y3	4588048,25
x4	313573,75
y4	4588048,25

**Taula 8** Taula del càlcul de coordenades de la imatge.

- ✓ Registrar la imatge en Geomedia Professional 6.0 de la següent manera:
  - Crear un magatzem ACCESS nou amb la plantilla normal.mdt.
  - Inserir una imatge interactiva, això s'explica a [22].
  - Registrar la imatge, per tal d'associar els quatre punts que delimiten la imatge. El resultat d'aquesta inserció es pot veure a la figura 20.

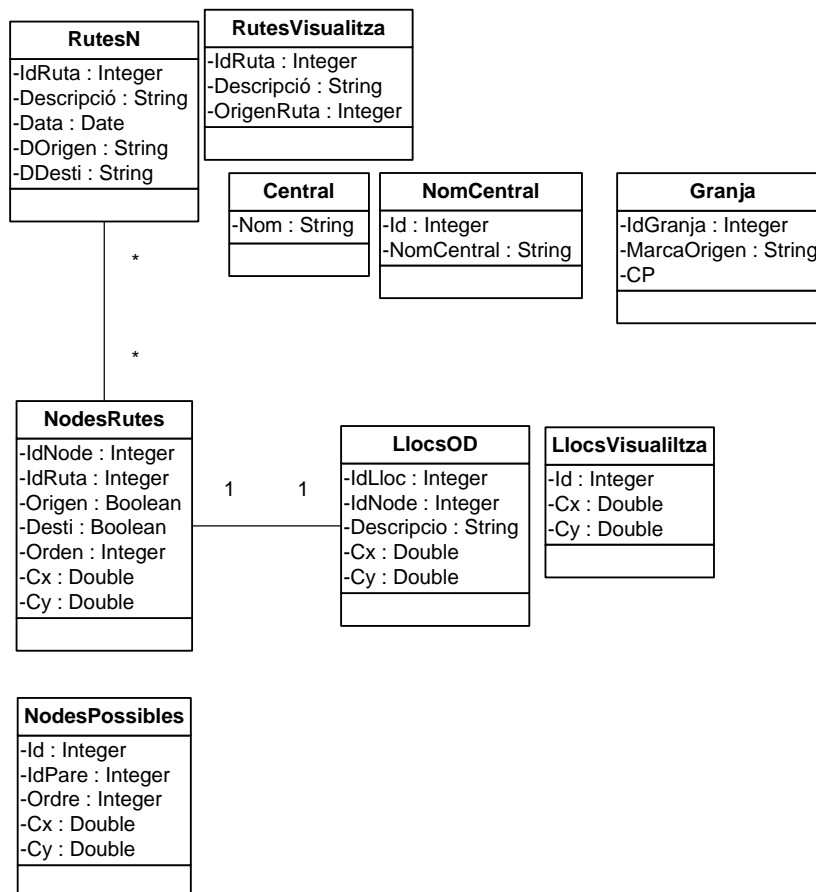


**Figura 20** Ortofoto inserida dins del SIG.

Una vegada definides les entitats que mostren la geografia de Catalunya i introduïda la imatge de la central, es té que pensar quin serà el model de dades que és necessitarà per resoldre aquest projecte.

### 6.2. Model de Dades

A continuació es mostra el diagrama del model de dades (veure figura 21) utilitzat en aquest projecte, aquest diagrama està creat mitjançant UML (**Unified Modeling Language**). Les entitats descrites en aquest diagrama són les que s'han utilitzat per implementar el sistema i poder implementar les seves necessitats, cada entitat és explicada més detalladament en la taula 9.



**Figura 21** Diagrama del model de dades en UML

<b>Entitat RutesN</b>	
<b>Descripció</b>	Emmagatzema les dades de totes les rutes que es van inserint en l'aplicació.
<b>Geometria</b>	De Línia
<b>Atributs</b>	IdRuta: Identifica a la Ruta. Clau primària Descripció: Fa una breu descripció de la ruta Data: Emmagatzema la data del dia que es va fer la ruta DOrigen: Descripció de l'Origen DDesti: Descripció del destí
<b>Entitat Granges</b>	
<b>Descripció</b>	Emmagatzema les situacions de les granges conegudes.
<b>Geometria</b>	De Punt
<b>Atributs</b>	IdGranja: Identifica la granja. Clau primària Marca oficial: Descripció de la granja CP: Codi postal
<b>Entitat RutesVisualitza</b>	
<b>Descripció</b>	Emmagatzema les dades de les rutes que es visualitzen.
<b>Geometria</b>	De Línia
<b>Atributs</b>	IdRuta: Identifica a la Ruta Descripció: Fa una breu descripció de la ruta OrigenRuta: Camp que ajudarà a filtrar en la llegenda els estils. (0=rutaVisualitzada;1=Ruta destí1;2=Ruta destí 2;3=Ruta destí 3)
<b>Entitat Central</b>	
<b>Descripció</b>	Emmagatzema les dades de la central del laboratori.
<b>Geometria</b>	De Punt
<b>Atributs</b>	Nom: Nom del Laboratori
<b>Entitat NomCentral</b>	
<b>Descripció</b>	Emmagatzema el nom de la central per mostrar al plànol.
<b>Geometria</b>	De Text
<b>Atributs</b>	NomCentral: Nom del Laboratori
<b>Entitat NodesRutes</b>	
<b>Descripció</b>	Emmagatzema tots els punts per on passa el cotxe en cada ruta en particular. També inclou els nodes de creuament, que una vegada identificats es guardaran com membres de les dues rutes a les que pertanyen.
<b>Geometria</b>	De Punt
<b>Atributs</b>	IdNode: Clau primària IdRuta: Identifica la ruta Origen: Indica si el node és un origen Destí: Indica si el node és un destí Ordre: Indica l'ordre del node dins de la ruta, per tal de conèixer la direcció. Cx: Coordenada x Cy: Coordenada y
<b>Entitat LlocsVisualitza</b>	
<b>Descripció</b>	Guarda els diferents destins en la planificació de les rutes.
<b>Geometria</b>	De Punt
<b>Atributs</b>	Id: Clau primària Cx: Coordenada x Cy: Coordenada y
<b>Entitat LlocsOD</b>	
<b>Descripció</b>	Descrui els Nodes que són origen o final.
<b>Geometria</b>	De Punt

<b>Atributs</b>	IdLloc: Identifica el lloc. IdNode: Identifica el node dins de la ruta Descripció: Descripció del lloc Cx: Coordenada x Cy: Coordenada y
-----------------	--

Taula 9 Descripció entitats sistema

### 6.3. Incorporació d'entitats en el *geoworkspace*

Una vegada dissenyat el model de dades, cal implementar aquestes entitats en el *geoworkspace* del projecte, per fer això cal seguir els següents passos:

- ✓ Definir un magatzem: Primer de tot es crearà un magatzem nou a partir del menú principal del Geomedia Professional 6.0 ("Almacen→Almacen Nuevo"), aquest magatzem serà de tipus Microsoft Access. S'ha escollit aquesta eina per la facilitat a l'hora d'aconseguir el programari, així com per l'experiència en ella.
- ✓ Definir una entitat: A continuació es crearà una entitat amb els atributs explicats anteriorment, com es pot observar en la figura 22. Aquest pas es farà des del menú principal ("Almacen→Definición de clase de entidad").

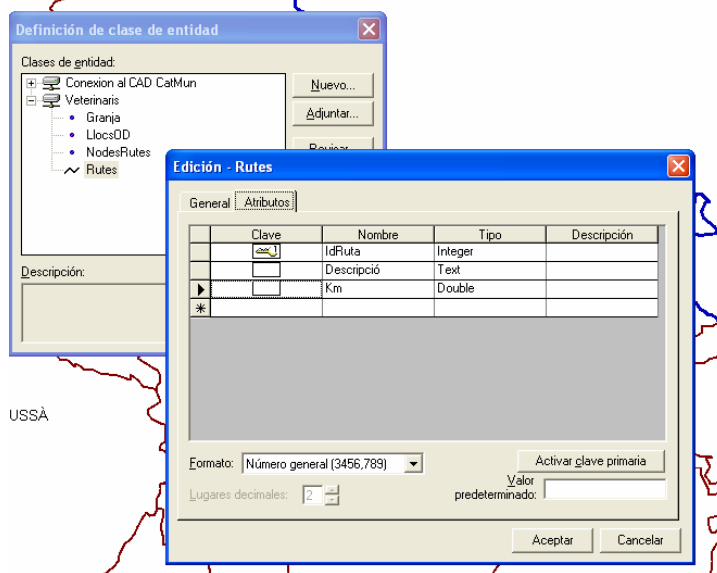


Figura 22 Definició d'entitats

Una vegada es tenen totes les entitats creades, es començarà amb la creació de les comandes que gestionaran aquestes. Aquestes comandes són explicades a continuació.



---

---

## 7. IMPLEMENTACIÓ DE COMANDES

---

---

En el capítol anterior s'ha definit el model de dades del SIG. En aquest capítol es fa la implementació de les funcions necessàries mitjançant comandes (veure apartat 4.4), tal i com s'explicarà en l'apartat 7.1.

En l'apartat 7.2 es farà un anàlisi de les funcionalitats del sistema, així com els actors que estan involucrats en aquestes.

Finalment s'explica en l'apartat 7.2.2 quines són les comandes implementades, la seva funcionalitat i les seves pantalles, així com es mostrarà el codi font més significatiu.

### 7.1. Requeriments

A continuació es mostrarà els requeriments que s'han fet servir per poder resoldre la implementació de les comandes:

- ✓ Les comandes implementades són: Entrada d'una nova ruta, recuperar ruta per data, recuperar ruta per origen i destí, introducció de granges conegudes, identificació de creuaments i planificació de rutes.
- ✓ La llegenda del *geoworkspace* creat només mostrarà les entitats : Granges, RutesVisualitza i LlocsVisualitza del model de dades creat. En el cas de l'entitat RutesVisualitza aquesta estarà categoritzada per 4 valors unitaris del camp OrigenRuta:
  - OrigenRuta=0: identifica les rutes que són visualitzades des de les comandes: entrada d'una nova ruta, recuperar ruta per data i recuperar ruta per origen i destí.
  - OrigenRuta=1: identifica la ruta que descriu el recorregut de la ruta planificada des d'un origen al destí 1 escollit per l'usuari.
  - OrigenRuta=2: identifica la ruta que descriu el recorregut de la ruta

planificada des del destí 1 al destí 2 escollit per l'usuari.

- OrigenRuta=3: identifica la ruta que descriu el recorregut de la ruta planificada des del destí 2 al destí 3 escollit per l'usuari.
- ✓ El fitxer d'entrada de les rutes està implementat per un programari extern, el qual filtra les dades que no necessita el sistema i prepara el fitxer pla perquè el sistema central les tracti. El fitxer té les següents característiques:
  - El format és el següent:

CoordenadaX;CoordenadaY;CoordenadaZ

- Les dades són emmagatzemades pel sistema cada 10 segons, opció que permet tenir una descripció de la ruta bastant acurada.
- Es defineix que la primera línia de cada fitxer serà considerada l'origen de la ruta i que la última línia del mateix serà considerada el destí, ja que el programari extern que crea el fitxer així ho descriu.
- La central del sistema està posicionada a les coordenades UTM que es mostren a la taula 10. Aquestes coordenades seran utilitzades com origen en les rutes iniciades en la central i com origen en la planificació de recorreguts.

X	31496611
Y	458284748
Situació	Juncosa (GARRIGUES)

**Taula 10** Coordenades Central

- ✓ El fitxer d'entrada de granges conegudes té el següent format:

MarcaOfical;CodiPostal;CoordenadaX;CoordenadaY

- ✓ Les rutes introduïdes són de doble sentit, per tant a l'hora de cercar rutes es pot escollir en una direcció o en una altra.

A continuació és farà un anàlisi de les comandes, indicant quins actors actuen en elles i fent una descripció acurada de totes les comandes implementades.

## 7.2. Anàlisis

En aquest capítol es farà un anàlisis del sistema dissenyat, explicant quins són els actors, així com totes les comandes que implementa. Totes les comandes s'han realitzat mitjançant el component de Geomedia Professional 6.0 [27] per a Visual Basic 6.0 [28].

### 7.2.1. Actors

Els actors que intervenen són:

- **Central:** Seu del laboratori, des d'on sortiran els cotxes en direcció les granges.
- **Destins:** Destins de les rutes que es fan.
- **Supervisor:** Usuari que utilitzarà l'aplicació creada.

### 7.2.2. Comandes

A continuació és mostrarà la informació que descriurà una comanda:

- **Resum:** Es farà una breu descripció de la utilitat de la comanda, dins d'aquest apartat s'explicaran els següents ítems:
  - **Funcionalitat:** Descriurà la funció de la comanda.
  - **Actors:** Indicarà quins són els actors que intervenen en la comanda.
  - **Condicions d'inici:** Condicions prèvies que es tenen que complir.
  - **Entitats:** Indicarà quines són les entitats involucrades en cada comanda.
  - **Errors:** Errors que es poden produir en l'execució de la comanda.
  - **Entrades:** Indica la informació d'entrada que és necessària.
  - **Sortides:** Indica la informació de sortida.
- **Funcionament:** En aquest apartat es descriurà les pantalles utilitzades i s'explicarà com l'usuari ha d'utilitzar les comandes.

- **Codi Font:** Mostrarà el codi principal utilitzat, amb breus comentaris.

Seguidament s'explicarà cada una de les comandes implementades.

### 7.2.2.1. Comanda "Entrada d'una nova Ruta"

Aquesta comanda és la comanda principal, ja que és la base per poder introduir rutes al sistema. La finalitat d'aquesta comanda és importar les coordenades incloses en un fitxer pla al sistema, i fer que la ruta importada aparegui en la pantalla en la posició exacta.

Per tal de fer això un programa extern al sistema captura totes les coordenades del GPS i les emmagatzema en un fitxer pla amb l'estructura definida a l'apartat 7.1. Aquest programa ja en retorna el fitxer amb les coordenades UTM necessàries per al sistema.

A continuació es mostra un resum d'aquesta comanda.

#### 7.2.2.1.1. Resum

A continuació en la taula 11, es mostra un breu resum de les característiques de la comanda.

<i>Descripció</i>	<i>Explicació</i>
<b>Funcionalitat</b>	Aquesta comanda té com objectiu crear una nova Ruta en el sistema a partir d'un fitxer pla amb les coordenades i de la informació que el supervisor indiqui.
<b>Actors</b>	Supervisor, Central, Destins.
<b>Condicions d'inici</b>	Existeix un fitxer amb la ruta emmagatzemada.
<b>Entitats</b>	RutesN: conté la nova ruta amb una geometria de línia. Nodesrutes: conté cada node de la ruta, especificant si és origen o destí. LlocsOD: conté els orígens i/o destins de les rutes. RutesVisualitza: conté la ruta introduïda per tal de visualitzar-la.
<b>Errors</b>	No omplir tota la informació demanada.
<b>Entrades</b>	Les dades d'aquesta ruta seran proporcionades mitjançant un fitxer pla amb la estructura explicada en l'apartat 7.1.
<b>Sortides</b>	Ubicació geogràfica de la ruta.

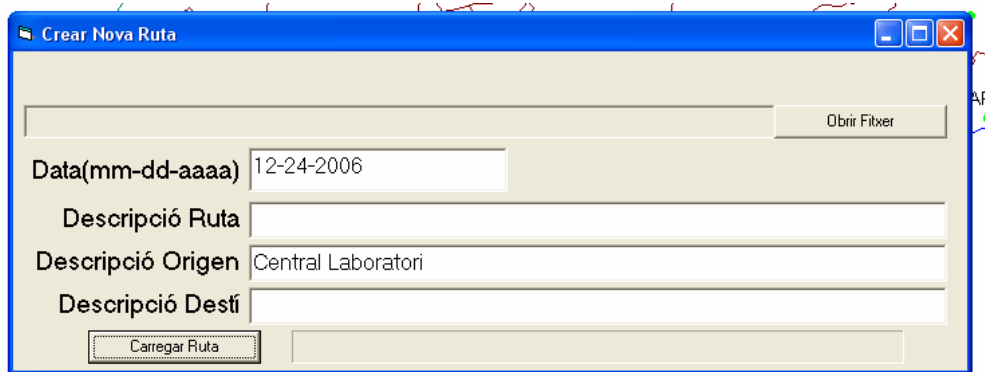
Taula 11 Resum de la comanda "Entrada d'una nova Ruta"

A continuació s'explicarà el funcionament d'aquesta comanda i les parts de la seva pantalla.

### 7.2.2.1.2. Funcionament

La pantalla (veure figura 23) d'aquesta comanda té els següents camps:

- ✓ Camí del fitxer d'entrada.
- ✓ Obrir Fitxer. Funció que obre la pantalla estàndard de Microsoft Windows per obrir un fitxer.
- ✓ Data. Data de realització de la ruta en format (mm-dd-aaaa).
- ✓ Descripció de la ruta. Descripció de la ruta que s'introduirà al sistema
- ✓ Descripció de l'origen. Nom del lloc on comença la ruta.
- ✓ Descripció del destí. Nom del lloc on finalitza la ruta.
- ✓ Carregar Ruta. Funció que introdueix la ruta en el sistema.
- ✓ Barra de progrés. Barra que mostra el progrés de la carrega del fitxer.

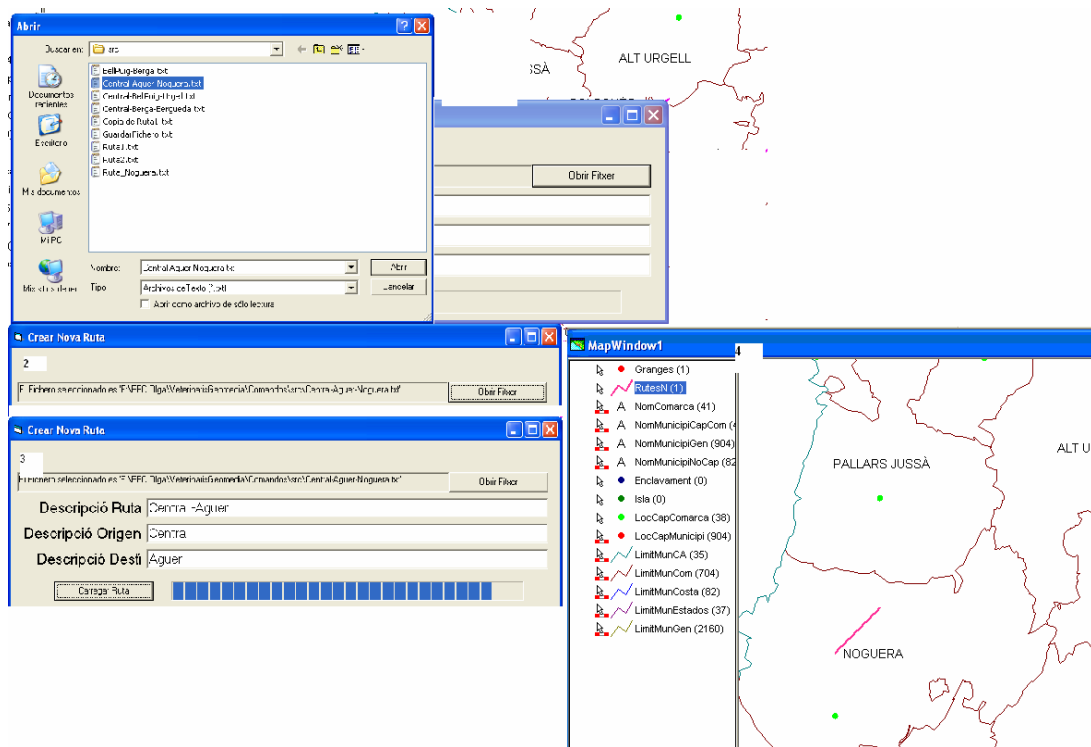


**Figura 23** Pantalla de la comanda "Entrada d'una nova Ruta"

El funcionament d'aquesta comanda és el següent (veure figura 24):

1. El supervisor escull el fitxer on està emmagatzemada la ruta realitzada. El sistema li retorna el camí que ha escollit.
2. Els supervisor omple les següents dades:
  - a. Data realització ruta.

- b. Descripció de la Ruta.
  - c. Descripció de l'origen.
  - d. Descripció del destí.
3. El supervisor pressiona "Carregar Ruta". El sistema llegeix el fitxer i crea en el sistema la ruta amb la data introduïda, introduint un registre en l'entitat RutesN amb la línia que descriu la ruta, i tants registres com nodes (punts) en l'entitat NodesRutes. De la mateixa manera guarda la descripció de l'origen i del destí en l'entitat LlocsOD. Per tal de no fer pensar a l'usuari que el sistema s'ha penjat, la pantalla indica l'estat de la comanda en una barra de progrés.
4. El sistema mostra la ruta en el programari de Geomedia Professional 6.0.



**Figura 24** Funcionament de la comanda "Entrada d'una nova Ruta".

A continuació es mostra el codi font més significatiu d'aquesta comanda.

### 7.2.2.1.3. Codi Font

---

```
'VARIABLES GLOBALES
'Variable que controla la connexió
Dim objConn As Connection
'Variable que permet crear una entitat de Tipus Línia
Dim PLine As PolylineGeometry
'Variables que controla l'entitat NodesRutes
Dim objRS_NodesRutes As GRecordset, objOP_NodesRutes As OriginatingPipe
'Variables que controla l'entitat RutesN
Dim objRS_Rutes As GRecordset, objOP_Rutes As OriginatingPipe
Dim objRS_RutesN As GRecordset, objOP_RutesN As OriginatingPipe
'Variables que controla l'entitat RutesVisualitza
Dim objRS_RutesV As GRecordset, objOP_RutesV As OriginatingPipe
Dim NumRuta As Integer
```

---

Primer de tot s'inicialitza la connexió a la base de dades del sistema, aquesta connexió serà igual per totes les comandes. També s'inicialitza l'objecte *PolylineGeometry* que emmagatzemarà la ruta importada.

---

#### Private Sub Form\_Load()

```
'Connexió a la Base de Dades
Set objConn = gobjGeoApp.Document.Connections(2)
'Creació de l'objecte que controla una línia, aquest objecte es diu PolylineGeometry
Set PLine = gobjGeoApp.CreateService("GeoMedia.PolylineGeometry")
'S'inicia la data del text amb la data del sistema, l'usuari podrà canviar aquesta.
Data.Text = Date$
```

#### End Sub

---

L'acció principal es desenvolupa en la funció *Cargar\_Ruta\_Click()*.

---

#### Private Sub Cargar\_Ruta\_Click()

```
Dim Cadena As String
Dim GeomOut As Variant
Dim GeomObj As Object
Dim GSS As Variant
Dim numF As Integer
Dim LineaNodo As String
Dim ValorX, ValorY, ValorZ As Double
Dim OrdenRuta As Integer
Dim PPoint As PointGeometry
'Comprovació de que totes les caixes de text estan omplertes
If FicheroAbierto.Caption = "" Or D_Ruta.Text = "" Or D_Origen.Text = "" Or D_Final.Text = "" Then
    MsgBox ("S'han d'introduir tots els camps")
Else
    'Creació de l'objecte que controla la transformació de Geometry a Storage
    Set GSS = gobjGeoApp.CreateService("Geomedia.GeometryStorageService")
```

---

---

```

'Objecte que controla una geometria de tipus punt
Set PPoint = gobjGeoApp.CreateService("GeoMedia.PointGeometry")

objConn.CreateOriginatingPipe objOP_NodesRutes
objOP_NodesRutes.Table = "NodesRutes"
Set objRS_NodesRutes = objOP_NodesRutes.OutputRecordset
Set objOP_NodesRutes = Nothing
numF = FreeFile
OrdenRuta = 1 'Ordena en l'ordre d'entrada els nodes
If Cuadro.FileName <> "" Then
  Call Crear_Reg_Rutes
  Open Cuadro.FileName For Input As numF
  Barra.Max = LOF(numF)
  While Not EOF(numF)
'Llegeix una línia del fitxer que representa un node i es guarda en el recordset vinculat a l'entitat NodesRutes
    Line Input #numF, LineaNodo
    If objRS_NodesRutes.Updatable Then
      objRS_NodesRutes.AddNew
      objRS_NodesRutes.GFields("IdRuta") = objRS_Rutes.GFields("IdRuta")
      objRS_NodesRutes.GFields("Orden") = OrdenRuta
      objRS_NodesRutes.GFields("Destino") = False
      PPoint.Origin.x = Val(Mid(LineaNodo, 1, InStr(1, LineaNodo, ";")))
      PPoint.Origin.y = Val(Right(LineaNodo, Len(LineaNodo) - InStr(InStr(1, LineaNodo, ";"), LineaNodo, ";")))
      PPoint.Origin.z = Val(Right(LineaNodo, Len(LineaNodo) - InStr(InStr(1, LineaNodo, ";") + 1, LineaNodo, ";")))
      GSS.GeometryToStorage PPoint, GeomOut
      objRS_NodesRutes.GFields("Geometry").Value = GeomOut
'Per cada node que introduïm aquesta funció ho annexa a la línia general
      Call Crear_Linea_Ruta(PPoint.Origin.x, PPoint.Origin.y, PPoint.Origin.z)
      'Si és la primera línia és l'origen
      If OrdenRuta = 1 Then
        objRS_NodesRutes.GFields("Origen") = True
      Else
        objRS_NodesRutes.GFields("Origen") = False
      End If
      objRS_NodesRutes.Update
      objRS_NodesRutes.MoveLast
      'Es guarda la informació de l'origen a l'entitat LlocsOD
      If OrdenRuta = 1 Then
        Call CrearLloc("O", objRS_NodesRutes.GFields("IdNode"), objRS_NodesRutes.GFields("Geometry").Value)
      End If
    End If
    OrdenRuta = OrdenRuta + 1
    Barra.Value = Barra.Value + Len(LineaNodo)
  Wend 'Això es farà mentre quedin línies per llegir del fitxer

  objRS_NodesRutes.MoveLast
  'Al últim introduït serà el destí
  If (objRS_NodesRutes.GFields("Orden") = OrdenRuta - 1) Then
    objRS_NodesRutes.Edit
    objRS_NodesRutes.GFields("Destino") = True
    objRS_NodesRutes.Update
    'Guarda la informació del destí a l'entitat LlocsOD
    Call CrearLloc("D", objRS_NodesRutes.GFields("IdNode"), objRS_NodesRutes.GFields("Geometry").Value)
  End If
'Una vegada llegit tot el fitxer, es guarda la línia creada a l'entitat RutesN a través de l'objecte
'Geomedia.GeometryStorageService que serveix per passar un objecte de tipus geometria a un tipus
'emmagatzematge i viceversa.

```

---



```
GSS.GeometryToStorage PLine, GeomOut
objRS_Rutes.MoveLast
objRS_Rutes.Edit
objRS_Rutes.GFields("Geometry").Value = GeomOut
objRS_Rutes.Update
```

```
Set objRS_NodesRutes = Nothing
Set objRS_Granjes = Nothing
Call Refrescar
MsgBox ("La Ruta ha estat introduïda")
```

```
Else
MsgBox ("Ha d'escollir un fitxer.")
End If
```

```
End If
Barra.Value = 0
```

'Es visualitza al sistema la ruta introduïda, afegint aquesta ruta a l'entitat RutesVisualitza que és la que es mostra a la llegenda del Geomedia Professional 6.0

```
Call Cargar_Visualitzar(NumRuta)
```

**End Sub**

#### **Private Function Borrar()**

'Esborra les rutes abans visualitzades per tal de no confondre a l'usuari

```
objConn.CreateOriginatingPipe objOP_RutesVB
objOP_RutesVB.Table = "RutesVisualitza"
Set objRS_RutesVB = objOP_RutesVB.OutputRecordset
If objRS_RutesVB.RecordCount <> 0 Then
objRS_RutesVB.MoveFirst
While Not objRS_RutesVB.EOF
objRS_RutesVB.Delete
objRS_RutesVB.MoveNext
Wend
End If
Call Refrescar
```

**End Function**

#### **Private Function CrearLloc(Tipus As String, Node As Integer, Geom As Variant)**

'Aquesta funció crea un registre en l'entitat LlocsOD amb les dades introduïdes per l'usuari

```
Dim objRS_Llocs As GRecordset, objOP_Llocs As OriginatingPipe
```

```
objConn.CreateOriginatingPipe objOP_Llocs
objOP_Llocs.Table = "LlocsOD"
Set objRS_Llocs = objOP_Llocs.OutputRecordset
objRS_Llocs.AddNew
objRS_Llocs.GFields("IdNode") = Node
objRS_Llocs.GFields("Geometry").Value = Geom
If Tipus = "O" Then
objRS_Llocs.GFields("Descripcio") = D_Origen.Text
Else
objRS_Llocs.GFields("Descripcio") = D_Final.Text
End If
```

```
objRS_Llocs.Update
```

**End Function**

#### **Private Function Crear\_Linea\_Ruta(x As Double, y As Double, z As Double)**

'Funció que adjunta un nou punt al total de la línia

```
Dim TPoint As Point
```

---

```
Set TPoint = gobjGeoApp.CreateService("GeoMedia.Point")
TPoint.x = x
TPoint.y = y
TPoint.z = z
PLine.Points.Add TPoint
Set TPoint = Nothing
```

**End Function**

#### **Private Sub Crear\_Reg\_Rutes()**

*'Funció que crea un registre a l'entitat de RutesN*

```
objConn.CreateOriginatingPipe objOP_Rutes
objOP_Rutes.Table = "RutesN"
Set objRS_Rutes = objOP_Rutes.OutputRecordset
If objRS_Rutes.Updatable Then
    objRS_Rutes.AddNew
    objRS_Rutes.GFields("Descripcio") = D_Ruta.Text
    objRS_Rutes.GFields("Data").Value = Data.Text
    objRS_Rutes.GFields("DOrigen").Value = D_Origen.Text
    objRS_Rutes.GFields("DDesti").Value = D_Final.text
    objRS_Rutes.Update
    objRS_Rutes.MoveLast
End If
NumRuta = objRS_Rutes.GFields("IdRuta")
```

**End Sub**

#### **Private Function Refrescar()**

*'Funció que refresca la pantalla del Geomedia Professional 6.0*

```
Dim LE As Object
gobjGeoApp.Document.Connections(2). BroadcastDatabaseChanges
For Each LE In gobjGeoApp.ActiveWindow.MapView.Legend.LegendEntries
    LE.LoadData
Next LE
gobjGeoApp.ActiveWindow.MapView.Legend.Refresh
gobjGeoApp.ActiveWindow.MapView.Refresh True
```

**End Function**

---

### **7.2.2.2. Comanda "Recuperació de recorreguts per data"**

Aquesta comanda recupera les rutes realitzades en una data concreta, i dóna a l'usuari la possibilitat d'escollir quina és la ruta que vol visualitzar. Per tal de poder realitzar això es fa una recerca de totes les rutes emmagatzemades en l'entitat RutesN i es filtra per la data que l'usuari hagi seleccionat en el calendari. Una vegada filtrat és mostra totes les rutes localitzades.

#### **7.2.2.2.1. Resum**

A continuació en la taula 12, es mostra un breu resum de les característiques de la comanda.

<i>Descripció</i>	<i>Explicació</i>
<b>Funcionalitat</b>	Aquesta comanda té com objectiu cercar les rutes realitzades en una data concreta.
<b>Actors</b>	Supervisor
<b>Condicions d'inici</b>	Existeixen rutes a l'entitat RutesN.
<b>Entitats</b>	RutesN: Entitat que té totes les rutes realitzades. RutesVisualitza: Entitat que tindrà la ruta que l'usuari vol visualitzar.
<b>Error</b>	No trobar rutes per la data seleccionada.
<b>Entrades</b>	Data.
<b>Sortides</b>	Llista de rutes d'una data en concret i Visualització d'aquestes.

**Taula 12** Resum de la comanda "Recuperar per Data"

#### 7.2.2.2.2. Funcionament

La pantalla (veure imatge (a) de la figura 25) d'aquesta comanda té els següents camps:

- ✓ Calendari. Objecte que permet a l'usuari indicar quina data vol consultar.
- ✓ Buscar Ruta Dia. Funció que troba les rutes emmagatzemades en la data indicada.
- ✓ Llista de Rutes. Llistat de les rutes emmagatzemades amb la data indicada per l'usuari.

El funcionament d'aquesta comanda és el següent:

1. El supervisor selecciona una data en el calendari de la pantalla.
2. El supervisor pressiona el botó "Buscar Ruta Data".
3. El sistema busca totes les rutes que es varen realitzar en la data demanada, en el cas de no trobar cap ruta mostrarà a l'usuari un missatge d'error (veure imatge (b) de la figura 25).
4. L'usuari fa doble clic damunt d'una de les rutes trobades, per tal de visualitzar-la, l'entitat encarregada de fer això és RutesVisualitza (veure figura 25).

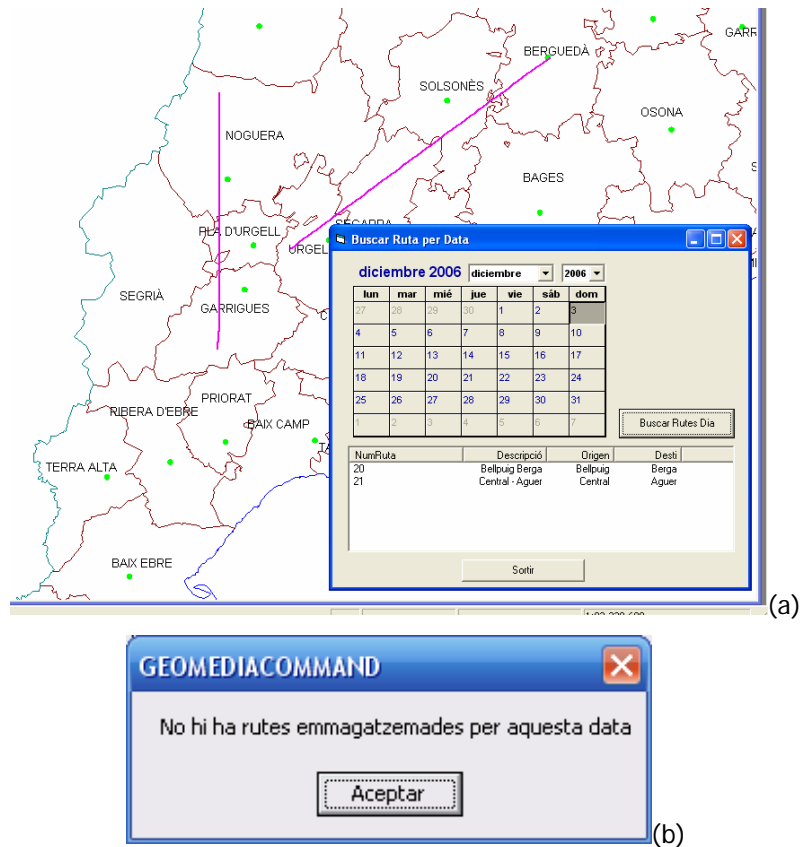


Figura 25 Pantalla de la comanda "Recuperació de recorreguts per data".

A continuació es mostra el codi font més significatiu d'aquesta comanda.

### 7.2.2.2.3. Codi

L'acció principal es desenvolupa en la funció `BuscarRuta_Click()`.

---

Option Explicit

Dim objConn As Connection

Dim objRS\_RutesN As GRecordset, objOP\_RutesN As OriginatingPipe

Dim objRS\_RutesV As GRecordset, objOP\_RutesV As OriginatingPipe

Dim objRS\_RutesVB As GRecordset, objOP\_RutesVB As OriginatingPipe

#### Private Sub BuscarRuta\_Click()

'Funció que acciona la recerca de les rutes existents en l'entitat `rutesN` amb la data introduïda per l'usuari, una vegada trobades són posicionades en un control `ListView` per tal de ser mostrades a l'usuari.

objConn.CreateOriginatingPipe objOP\_RutesV

objOP\_RutesV.Table = "RutesVisualitza"

Set objRS\_RutesV = objOP\_RutesV.OutputRecordset

Dim i As Integer

---

objConn.CreateOriginatingPipe objOP\_RutesN

---

```

objOP_RutesN.Table = "RutesN"
MsgBox (Calendario.Value)
'Es filtra per la data que l'usuari a seleccionat en el calendari.
objOP_RutesN.Filter = "Data=#" & Calendario.Value & "#"
Set objRS_RutesN = objOP_RutesN.OutputRecordset
If objRS_RutesN.RecordCount <> 0 Then
    Origenes.ColumnHeaders.Clear
    Origenes.ColumnHeaders.Add(1, , "NumRuta", 1620).Tag = "Número"
    Origenes.ColumnHeaders.Add(2, , "Descripció", 1110, lvwColumnRight).Tag = "Texto"
    Origenes.ColumnHeaders.Add(3, , "Origen", 840, lvwColumnRight).Tag = "Texto"
    Origenes.ColumnHeaders.Add(4, , "Destí", 840, lvwColumnRight).Tag = "Texto"
    i = 1
    objRS_RutesN.MoveFirst
    While Not objRS_RutesN.EOF
        With Origenes.ListItems.Add(i, , objRS_RutesN.GFields("idRuta").Value)
            .SubItems(1) = objRS_RutesN.GFields("Descripcio").Value
            .SubItems(2) = objRS_RutesN.GFields("DOrigen").Value
            .SubItems(3) = objRS_RutesN.GFields("DDesti").Value
        End With
        objRS_RutesN.MoveNext
        i = i + 1
    Wend
End If
End Sub

```

#### Private Sub Origenes\_Db1Click()

```

'Funció que visualitza la ruta seleccionada al fer doble clic damunt de l'identificador de ruta
objConn.CreateOriginatingPipe objOP_RutesV
objOP_RutesV.Table = "RutesVisualitza"
Set objRS_RutesV = objOP_RutesV.OutputRecordset

Dim NumRuta As String
NumRuta = Origenes.SelectedItem.Text
MsgBox (NumRuta)
objConn.CreateOriginatingPipe objOP_RutesN
objOP_RutesN.Table = "RutesN"
objOP_RutesN.Filter = "IdRuta = " & NumRuta
Set objRS_RutesN = objOP_RutesN.OutputRecordset
' Es passa el registre de l'entitat RutesN a l'entitat RutesVisualitza amb el atribut OrigenRuta igual
'a 0, per tal que el sistema filtri el estil amb el que té que mostrar la ruta.
While Not objRS_RutesN.EOF
    If objRS_RutesV.Updatable Then
        objRS_RutesV.AddNew
        objRS_RutesV.GFields("Descripcio") = objRS_RutesN.GFields("Descripcio")
        objRS_RutesV.GFields("Geometry").Value = objRS_RutesN.GFields("Geometry").Value
        objRS_RutesV.GFields("OrigenRuta").Value = 0
        objRS_RutesV.Update
    End If
    objRS_RutesN.MoveNext
Wend
Call Refrescar

```

**End Sub**

**7.2.2.3. Comanda “Recuperació de rutes per origen i destí”**

Aquesta comanda recupera les rutes inserides en tot el sistema, donant com informació quin és l'origen i el destí. Dóna a l'usuari la possibilitat d'escollir quina és la ruta que vol visualitzar. Per tal de poder realitzar això es fa una recerca de totes les rutes emmagatzemades en l'entitat “RutesN” i les mostra en un llistat.

**7.2.2.3.1. Resum**

A continuació en la taula 13, es mostra un breu resum de les característiques de la comanda:

<i>Descripció</i>	<i>Explicació</i>
<b>Funcionalitat</b>	Aquesta comanda té com objectiu cercar les rutes i mostrar-les indicant el seu origen i destí.
<b>Actors</b>	Supervisor.
<b>Condicions d'inici</b>	Existeixen rutes a l'entitat RutesN.
<b>Entitats</b>	RutesN: entitat que té totes les rutes realitzades. RutesVisualitza: entitat que tindrà la ruta que l'usuari vol visualitzar.
<b>Errors</b>	No trobar rutes.
<b>Entrades</b>	
<b>Sortides</b>	Llista de rutes i visualització d'aquestes

**Taula 13** Resum de la comanda “Recuperar per Origen i Destí”

**7.2.2.3.2. Funcionament**

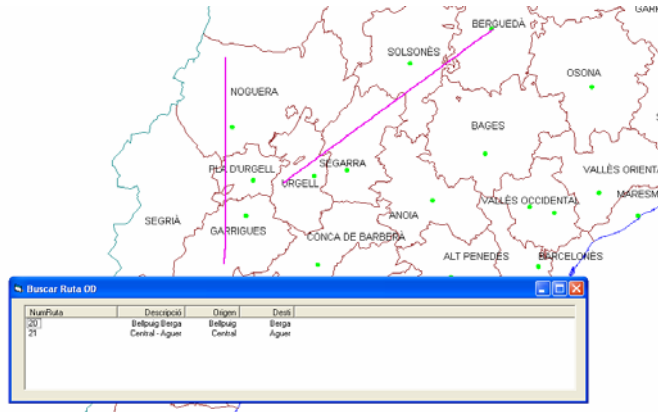
La pantalla (veure figura 26) d'aquesta comanda té els següents camps:

- ✓ Llista de Rutes. Llistat de totes les rutes emmagatzemades

El funcionament d'aquesta comanda és el següent:

1. El sistema carrega a l'inici totes les rutes que es tenen emmagatzemades mostrant la informació del seu origen i el seu destí, així com la descripció que l'usuari va indicar.
2. L'usuari fa doble clic damunt d'una de les rutes trobades, per tal de visualitzar-la. L'entitat encarregada d'això és l'entitat “RutesVisualitza” ( veure figura 26). En el cas de no trobar

cap ruta el llistat sortirà en blanc.



**Figura 26** Pantalla i funcionament de la comanda “Recuperació de rutes per Origen i Destí”

### 7.2.2.3.3. Codi

A continuació es mostra el codi font més significatiu d'aquesta comanda. El codi per mostrar la ruta seleccionada és el mateix que en la funció `Origens_DbClick()` de la comanda anterior.

---

Option Explicit

Dim objConn As Connection

Dim objRS\_Origens As GRecordset, objOP\_Origens As OriginatingPipe

Dim objRS\_Destins As GRecordset, objOP\_Destins As OriginatingPipe

Dim objRS\_RutesV As GRecordset, objOP\_RutesV As OriginatingPipe

Dim objRS\_RutesN As GRecordset, objOP\_RutesN As OriginatingPipe

Dim objRS\_RutesVB As GRecordset, objOP\_RutesVB As OriginatingPipe

#### Private Sub Form\_Load()

'Aquesta comanda es carrega des del principi. Carrega totes les rutes que té emmagatzemades diferenciant l'origen i el destí

Dim I As Integer

Set objConn = gobjGeoApp.Document.Connections(2)

objConn.CreateOriginatingPipe objOP\_Origens

objOP\_Origens.Table = "RutesN"

Set objRS\_Origens = objOP\_Origens.OutputRecordset

If objRS\_Origens.RecordCount <> 0 Then

Origens.ColumnHeaders.Clear

Origens.ColumnHeaders.Add(1, "NumRuta", 1620).Tag = "Número"

Origens.ColumnHeaders.Add(2, "Descripció", 1110, lvwColumnRight).Tag = "Textó"

Origens.ColumnHeaders.Add(3, "Origen", 840, lvwColumnRight).Tag = "Textó"

Origens.ColumnHeaders.Add(4, "Destí", 840, lvwColumnRight).Tag = "Textó"

I = 1

objRS\_Origens.MoveFirst

While Not objRS\_Origens.EOF

```

With Orgens.ListItems.Add(I, , objRS_Orgens.GFields("idRuta").Value)
    .SubItems(1) = objRS_Orgens.GFields("Descripcio").Value
    .SubItems(2) = objRS_Orgens.GFields("DOrigen").Value
    .SubItems(3) = objRS_Orgens.GFields("DDesti").Value
End With
objRS_Orgens.MoveNext
I = I + 1
Wend
End If
End Sub

```

#### 7.2.2.4. Comanda “Introducció de les granges conegudes”

Aquesta comanda insereix granges conegudes en el sistema, aquestes granges són els possibles clients del laboratori. Per entrar aquestes comandes es llegirà un fitxer amb el format descrit en el apartat 7.1, on cada línia del fitxer és una granja.

##### 7.2.2.4.1. Resum

A continuació en la taula 14, es mostra un breu resum de les característiques de la comanda:

<i>Descripció</i>	<i>Explicació</i>
<b>Funcionalitat</b>	Aquesta comanda permet inserir en el sistema una llista de granges georeferenciades per tal de poder utilitzar-les com futurs destins en l'aplicació.
<b>Actors</b>	Supervisor
<b>Condicions d'inici</b>	
<b>Entitats</b>	Granges
<b>Error</b>	
<b>Entrades</b>	Fitxer d'entrada
<b>Sortides</b>	Visualització de les granges.

**Taula 14** Resum de la comanda “Incorporar Granges”

##### 7.2.2.4.2. Funcionament

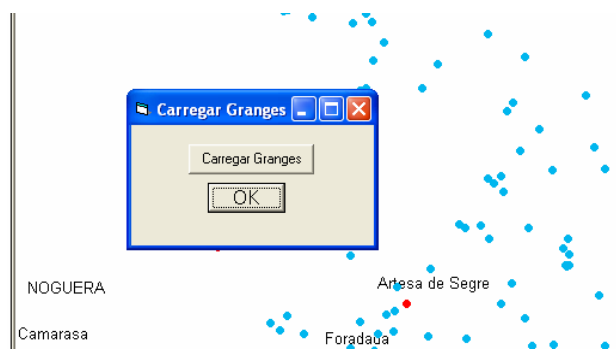
La pantalla(veure figura 27) d'aquesta comanda té els següents camps:

- ✓ Carregar Granges. Funció que importa un fitxer pla amb totes les granges conegudes l'entitat “Granges”.



El funcionament d'aquesta comanda és el següent:

1. El sistema llegirà totes les granges incloses en el fitxer d'entrada i les emmagatzemarà en l'entitat Granges.
2. Seguidament el sistema mostrarà les dades. (veure figura 27).



**Figura 27** Pantalla de la comanda “Introducció de les granges conegudes”

A continuació es mostra el codi font més significatiu d'aquesta comanda.

#### 7.2.2.4.3. Codi

---

Option Explicit

Dim objConn As Connection

Dim objRS\_Granjes As GRecordset, objOP\_Granjes As OriginatingPipe

#### Private Sub Command2\_Click()

Dim Cadena As String

Dim GeomOut As Variant

Dim GSS As Variant

Dim numF As Integer

Dim LineaNodo As String

Dim PPoint As PointGeometry

objConn.CreateOriginatingPipe objOP\_Granjes

objOP\_Granjes.Table = "Granges"

Set objRS\_Granjes = objOP\_Granjes.OutputRecordset

Set GSS = gobjGeoApp.CreateService("Geomedia.GeometryStorageService")

Set PPoint = gobjGeoApp.CreateService("GeoMedia.PointGeometry")

numF = FreeFile

Cuadro.Filter = "Archivos de Texto (\*.txt)|\*.txt"

Cuadro.ShowOpen

Open Cuadro.FileName For Input As numF

While Not EOF(numF)

' Es llegeix una línia del fitxer

Line Input #numF, LineaNodo

---

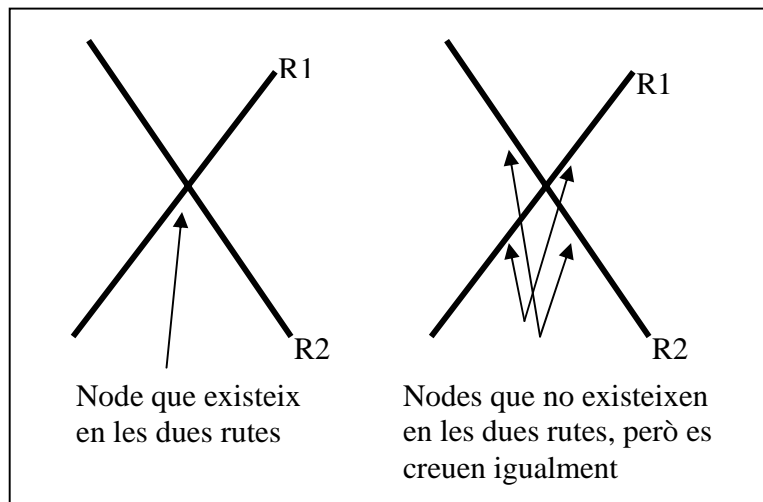
```

If objRS_Granjes.Updatable Then
  objRS_Granjes.AddNew
  'Es guarda cada atribut en el seu camp respectivament
  objRS_Granjes.GFields("MarcaOficial") = Mid(LineaNodo, 1, 6)
  objRS_Granjes.GFields("CP") = Mid(LineaNodo, 8, 5)
  'Es crea un Objecte de tipus PointGeometry
  PPoint.Origin.X = Val(Mid(LineaNodo, 14, 8))
  PPoint.Origin.Y = Val(Mid(LineaNodo, 23, 9))
  PPoint.Origin.Z = 0
  'Es passa d'un objecte de geometria a un objecte de emmagatzematge
  GSS.GeometryToStorage PPoint, GeomOut
  ' I es guarda en el camp corresponent
  objRS_Granjes.GFields("Geometry").Value = GeomOut
  objRS_Granjes.Update
End If
Wend
Set objRS_Granjes = Nothing
Call Refrescar
End Sub

```

### 7.2.2.5. Comanda "Identificació de creuaments"

Aquesta comanda té la finalitat de trobar tots els punts de creuament entre dues rutes. La primera manera que es pensa per solucionar aquesta comanda és buscar tots els nodes comuns entre dues rutes, però es troba que aquesta no és la solució més òptima, ja que es pot donar el cas de tenir algun punt en comú però que en canvi no estigui definit en cap de les dues rutes (veure figura 28) .



**Figura 28** Tipus de creuament entre rutes

La solució per aquest problema és crear una consulta espacial amb Geomedia Professional 6.0 (veure apartat 4.2). En particular s'utilitzarà el tipus de consulta "interseccions espacials" [27], aquesta retornarà totes les parts comuns que tenen dues entitats. Aquesta consulta es realitza amb un objecte de Geomedia Professional 6.0 , aquest objecte és diu *SpatialIntersectionPipe*.

Les consultes en Geomedia Professional 6.0 es fan en general entre dues entitats i es comparen tots els registres amb tots, però en aquest cas es filtrarà per l'identificador de la ruta que es vol consultar si té interseccions, ja que el que es vol consultar és la intersecció d'una ruta en particular amb totes les altres rutes introduïdes.

Una vegada feta la consulta i si s'ha trobat algun punt en comú, el següent pas és incloure aquest nou punts trobats a les seves respectives rutes, per tal de tenir cada vegada més definides aquestes. En aquest punt es troba un altre petit problema, tal i com s'ha explicat en l'apartat 7.2.2.1 l'entitat que emmagatzema els nodes de les rutes també emmagatzema l'ordre d'aquests nodes, és per això que a l'hora d'introduir el nou node, primer es tindrà que buscar quina és la seva posició correcta i moure una posició a tots els nodes que el segueixen. Solucionat aquest punt ja es té el nou node introduït i es podrà utilitzar en la recerca de camins que en l'apartat 7.2.2.6 s'explicarà amb més detall.

A continuació en la taula 15, es mostra un breu resum de les característiques de la comanda:

#### 7.2.2.5.1. Resum

<i>Descripció</i>	<i>Explicació</i>
<b>Funcionalitat</b>	Cercar tots els punts en comú d'una ruta en particular amb totes les rutes introduïdes anteriorment.
<b>Actors</b>	Supervisor
<b>Condicions d'inici</b>	Existència de rutes
<b>Entitats</b>	RutesN: entitat que emmagatzema les rutes a consultar NodesRutes: entitat on s'ha afegirà els nous nodes.
<b>Errors</b>	No hi hagin rutes introduïdes
<b>Entrades</b>	Cap
<b>Sortides</b>	Cap

Taula 15 Resum de la comanda "Identificació de creuaments"

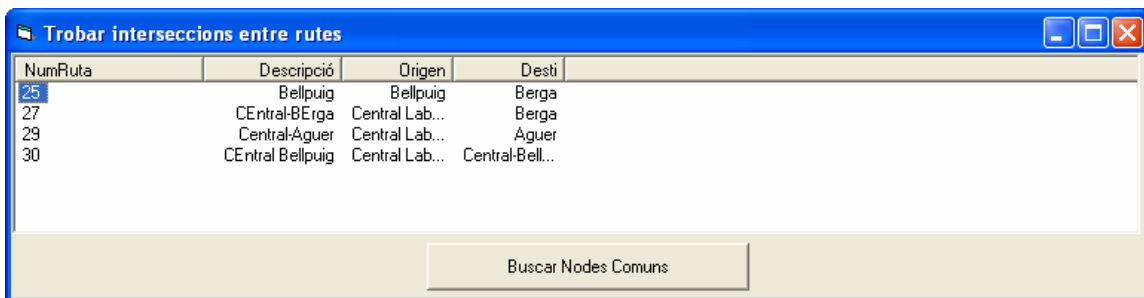
### 7.2.2.5.2. Funcionament

La pantalla(veure figura 29) d'aquesta comanda té els següents camps:

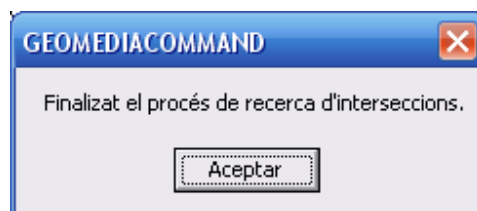
- ✓ NumRuta: identificador de la ruta.
- ✓ Descripció: descripció que l'usuari ha donat a la ruta en particular.
- ✓ Origen: origen que l'usuari ha identificat.
- ✓ Destí : destí que l'usuari ha identificat.
- ✓ Botó "Buscar Nodos Comuns": activa l'algoritme de recerca.

El funcionament d'aquesta comanda és el següent:

1. L'usuari escull la ruta que vol comparar amb les altres rutes per trobar els nodes de creuament.
2. L'usuari pressiona el botó "Buscar Nodos Comuns".
3. El sistema localitza els nodes de creuament i els afegeix a les dues rutes a les quals pertanyen, quan finalitza dona el missatge de la figura 30.



**Figura 29** Pantalla "Identificació de creuaments"



**Figura 30** Missatge de finalització de la comanda "Identificació de creuaments"

### 7.2.2.5.3. Codi

A continuació es mostrarà les dues funcions més importants del codi, la primera funció crea la consulta d'intersecció i després interpreta el resultat, la segona insereix el node trobat en les dues rutes a on pertany.

---

#### Private Sub Command4\_Click()

```
Dim NumRuta As String
Dim objOPipe As OriginatingPipe
'Aquest és l'objecte que fa la consulta d'intersecció espacial en Geomedia Professional 6.0
Dim objQry As New SpatialIntersectionPipe
Dim objLE As LegendEntry
NumRuta = Origen.SelectedItem.Text

'Comença la consulta
'La primera entitat de la consulta serà RutesN però filtrant que no sigui la ruta seleccionada
objConn.CreateOriginatingPipe objOPipe
With objOPipe
    .GeometryFieldName = "Geometry"
    .Table = "RutesN"
    .Filter = "idruta <>" & NumRuta
end With
' Assignació de l'entitat filtrada a la consulta
Set objQry.LeftRecordset = objOPipe.OutputRecordset
objQry.LeftGeometryFieldName = objOPipe.GeometryFieldName
Set objLE = Nothing
Set objOPipe = Nothing
'La segona entitat de la consulta serà RutesN però filtrant per la ruta seleccionada
objConn.CreateOriginatingPipe objOPipe
With objOPipe
    .GeometryFieldName = "Geometry"
    .Table = "RutesN"
    .Filter = "idruta =" & NumRuta
End With

' Assignació de l'entitat filtrada a la consulta
With objQry
    Set .RightRecordset = objOPipe.OutputRecordset
    .RightGeometryFieldName = objOPipe.GeometryFieldName
    .OutputGeometryFieldName = "OutputGeometry" 'Es defineix quin serà el nom del camp Geometry
    .SpatialOperator = gmsqTouches 'Tipus de consulta
    .OutputStatusFieldName = "status"
End With

Set objOPipe = Nothing
Dim R As GRecordset
Set R = objQry.OutputRecordset
```

---

'R serà el recordset on es guarda el *outputrecordset* que retorna, cada registre pot tenir una geometria diferent, ja que per exemple si troba un únic punt ho guarda com *OrientedPointGeometry* i si troba varies interseccions ho guarda com un *GeometryCollection* on cada ítem és un *OrientedPointGeometry*

```
Dim Gout As Object
Dim GSS1 As Variant
Dim n As Integer
Set GSS1 = gobjGeoApp.CreateService("Geomedia.GeometryStorageService")
R.MoveFirst
While Not R.EOF
    If R.GFields("idRuta") <> Val(NumRuta) Then
        MsgBox ("entra <>")
        'Es passa de tipus Storage a Geometry per saber quin tipus de dades és i treballar segons aquesta
        GSS1.StorageToGeometry R.GFields("OutputGeometry").Value, Gout
        If Gout.Type = "GeometryCollection" Then
            For n = 1 To Gout.Count 'Per tots els punts que hagi trobat
                'Funció que carrega a l'entitat NodesRutes el node trobat en les dues rutes on NumRuta és la ruta
                'seleccionada i R.GFields("idRuta") és la ruta on la consulta ha trobat mínim un punt en comú.
                Call CargarNodoEncontrado(Gout.Item(n), R.GFields("idruta"), NumRuta)
            Next
        End If
        If Gout.Type = "OrientedPointGeometry" Then 'Només a trobat un punt en comú
            Call CargarNodoEncontrado(Gout, R.GFields("idruta"), NumRuta)
        End If
    End If
    R.MoveNext
Wend
MsgBox ("Finalitzat el procés de recerca de nodes comuns.")
End Sub
```

**Private Function CargarNodoEncontrado(Nodo As Object, Ruta1 As String, Ruta2 As String)**

'Els paràmetres són, el objecte Point nou que ha trobat, i les dues rutes a les quals pertany.

```
Dim GeomOut As Variant
Dim GSS2 As Variant
Set GSS2 = gobjGeoApp.CreateService("Geomedia.GeometryStorageService")
'Primer es carrega el nou node en la ruta del segon paràmetre, sempre que no existeixi ja.
objConn.CreateOriginatingPipe objOP_NodesRutes
objOP_NodesRutes.Table = "NodesRutes"
objOP_NodesRutes.Filter = "idruta=" & Ruta1 & " and Cx=" &
Str(Val(Replace(Str(Nodo.Origin.X), ",", "."))) & " and Cy=" & Str(Val(Replace(Str(Nodo.Origin.Y),
",", "."))) & " and Cz=" & Str(Val(Replace(Str(Nodo.Origin.Z), ",", ".")))
Set objRS_NodesRutes = objOP_NodesRutes.OutputRecordset 'Es busca si existeix
If objRS_NodesRutes.RecordCount = 0 Then 'En el cas de que no estigui s'insereix.
    objConn.CreateOriginatingPipe objOP_NodesRutes2
    objOP_NodesRutes2.Table = "NodesRutes"
    'Es busca el punt on es té que introduir dins del ordre que té la ruta i es modifiquen tots els ordres
    'dels nodes que van darrere.
    objOP_NodesRutes2.Filter = "(idruta=" & Ruta1 & ") and (Cx>" &
```

```
Str(Val(Replace(Str(Nodo.Origin.X), ",", "."))) & ")"  
  Set objRS_NodesRutes2 = objOP_NodesRutes2.OutputRecordset  
  Dim OrdenPrimero As Integer  
  objRS_NodesRutes2.MoveFirst  
  OrdenPrimero = objRS_NodesRutes2.GFields("orden")  
  Dim MinOrden As Integer  
  MinOrden = OrdenPrimero  
  If objRS_NodesRutes2.Updatable Then  
    objRS_NodesRutes2.MoveFirst  
    While Not objRS_NodesRutes2.EOF  
      If MinOrden > objRS_NodesRutes2.GFields("Orden") Then  
        MinOrden = objRS_NodesRutes2.GFields("Orden")  
      End If  
      objRS_NodesRutes2.Edit  
      objRS_NodesRutes2.GFields("Orden") = objRS_NodesRutes2.GFields("Orden") + 1  
      objRS_NodesRutes2.Update  
      objRS_NodesRutes2.MoveNext  
    Wend  
    objRS_NodesRutes2.AddNew  
    objRS_NodesRutes2.GFields("IdRuta") = Ruta1  
    objRS_NodesRutes2.GFields("Orden") = MinOrden  
    objRS_NodesRutes2.GFields("Cx") = Nodo.Origin.X  
    objRS_NodesRutes2.GFields("Cy") = Nodo.Origin.Y  
    objRS_NodesRutes2.GFields("Cz") = Nodo.Origin.Z  
    GSS2.GeometryToStorage Nodo, GeomOut  
    objRS_NodesRutes2.GFields("Geometry").Value = GeomOut  
    objRS_NodesRutes2.Update  
  End If  
End If
```

'Es fa el mateix amb la altra ruta.

```
objConn.CreateOriginatingPipe objOP_NodesRutesr2  
objOP_NodesRutesr2.Table = "NodesRutes"  
objOP_NodesRutesr2.Filter = "idruta=" & Ruta2 & " and Cx=" &  
Str(Val(Replace(Str(Nodo.Origin.X), ",", "."))) & " and Cy=" & Str(Val(Replace(Str(Nodo.Origin.Y),  
",", "."))) & " and Cz=" & Str(Val(Replace(Str(Nodo.Origin.Z), ",", ".")))  
  Set objRS_NodesRutesr2 = objOP_NodesRutesr2.OutputRecordset
```

```
If objRS_NodesRutesr2.RecordCount = 0 Then  
  objConn.CreateOriginatingPipe objOP_NodesRutesr22  
  objOP_NodesRutesr22.Table = "NodesRutes"  
  objOP_NodesRutesr22.Filter = "idruta=" & Ruta2 & " and Cx>" &  
Str(Val(Replace(Str(Nodo.Origin.X), ",", ".")))  
  Set objRS_NodesRutesr22 = objOP_NodesRutesr22.OutputRecordset
```

```
objRS_NodesRutesr22.MoveFirst  
OrdenPrimero = objRS_NodesRutesr22.GFields("orden")  
MinOrden = OrdenPrimero  
If objRS_NodesRutesr22.Updatable Then  
  objRS_NodesRutesr22.MoveFirst  
  While Not objRS_NodesRutesr22.EOF
```

---

---

```
If MinOrden > objRS_NodesRutesr22.GFields("Orden") Then
    MinOrden = objRS_NodesRutesr22.GFields("Orden")
End If
objRS_NodesRutesr22.Edit
objRS_NodesRutesr22.GFields("Orden") = objRS_NodesRutesr22.GFields("Orden") + 1
objRS_NodesRutesr22.Update
objRS_NodesRutesr22.MoveNext
Wend
objRS_NodesRutesr22.AddNew
objRS_NodesRutesr22.GFields("IdRuta") = Ruta2
objRS_NodesRutesr22.GFields("Orden") = MinOrden 'OrdenPrimero
objRS_NodesRutesr22.GFields("Cx") = Nodo.Origin.X
objRS_NodesRutesr22.GFields("Cy") = Nodo.Origin.Y
objRS_NodesRutesr22.GFields("Cz") = Nodo.Origin.Z
GSS2.GeometryToStorage Nodo, GeomOut
objRS_NodesRutesr22.GFields("Geometry").Value = GeomOut
objRS_NodesRutesr22.Update
End If
End If
End Function
```

---

#### 7.2.2.6. Comanda "Planificació de rutes"

Aquesta comanda té la finalitat de buscar un camí entre la central del laboratori i els destins que l'usuari indiqui en l'ordre que indiqui. Aquests camins seran cercats des de la informació que es té de totes les rutes que s'han introduït en el sistema.

La comanda té la restricció de només poder cercar un camí amb tres destins diferents.

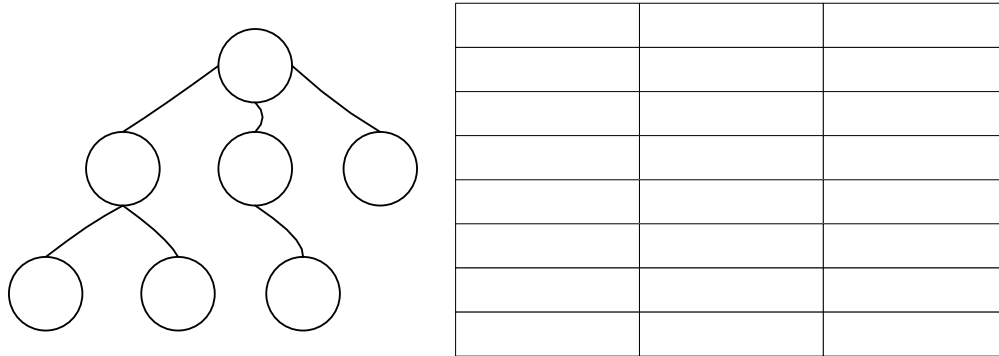
Per buscar un camí es farà servir l'algoritme d'amplada – primer explicat en l'apartat 5.2.2. Aquest algoritme està implementat de la següent manera:

- Llista de nodes possibles (LNP): Es té una entitat en la base de dades del sistema on es guarda tots els nodes possibles que es poden visitar, aquesta entitat té la següent estructura:
  - Coordenades x, y del node.
  - Identificador del node pare.
  - Identificador de l'ordre d'entrada.

Cada vegada que un node és afegit a la llista, l'hi és assignat un número d'ordre el qual indicarà en tot moment quins és el següent node a cercar. D'aquesta manera es compleix la norma de recerca del següent node de l'algoritme d'amplada-primer, en la figura 31 es



mostra gràficament el mètode seguit.



**Figura 31** Representació gràfica del mètode per omplir la LNP

- Mètode de recerca de següents: Per tal de saber quins nodes s'han d'afegir a la LNP des del node actual, és mira quins nodes són els següents on es pot anar des de l'actual, això es detecta per que des del node actual es busca els nodes anteriors i següents dins de les rutes on el node actual existeix. Una vegada es tenen els nodes possibles es controla que aquests nodes no estiguin en la LNP i en el cas d'existir no s'introdueixen. En el cas de que un d'aquests nodes ja sigui el destí es dóna per finalitzat l'algoritme de recerca, i es comença a crear el camí trobat.
- Mètode de cerca camí: Una vegada es té identificat el destí dins de la LNP, es fa el camí invers buscant des del node destí el seu node pare i així successivament fins que es troba el node origen. El node origen s'identifica per que tant l'identificador del node com el node pare són el mateix. D'aquesta manera es tindran tots els nodes que componen el camí trobat.

A continuació en la taula 16, es mostra un breu resum de les característiques de la comanda:

**7.2.2.6.1. Resum**

**B**

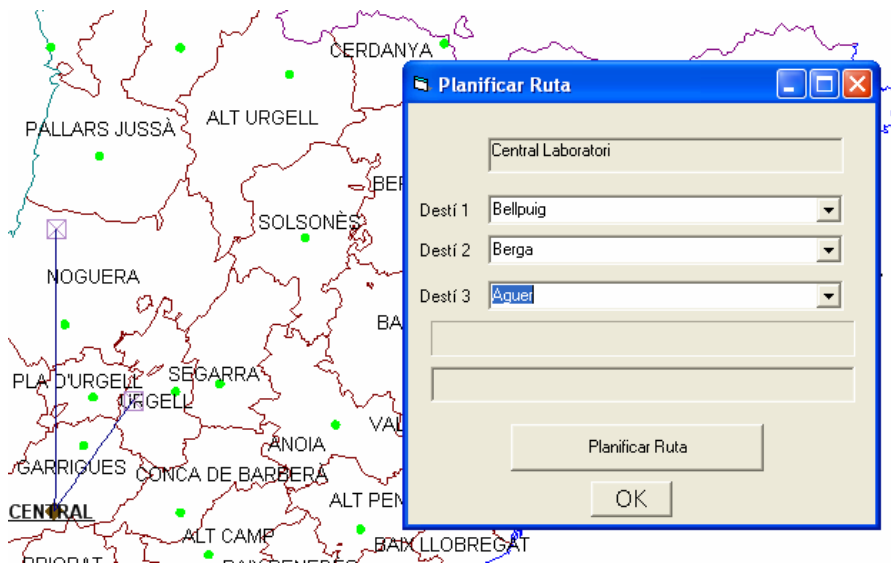
<i>Descripció</i>	<i>Explicació</i>
<b>Funcionalitat</b>	Cercar un camí des de la central del laboratori i tres destins, en l'ordre indicat per l'usuari.
<b>Actors</b>	Supervisor
<b>Condicions d'inici</b>	Tenir unes rutes introduïdes i mínim un destí.
<b>Entitats</b>	RutesNodes: en aquesta entitat es cerca tots els nodes possibles. NodesPosibles: en aquesta entitat es guarda els nodes amb els que es treballa. RutesVisualitza: en aquesta entitat es mostra la ruta resultant. Indicant a quin destí pertany. LlocsVisualitza: en aquesta entitat es mostra els destins.
<b>Errors</b>	
<b>Entrades</b>	Coordenades origen, coordenades destins.
<b>Sortides</b>	Ruta resultant.

**Taula 16** Resum de la comanda "Planificació de Rutes"

#### 7.2.2.6.2. Funcionament

La pantalla (veure figura 32) d'aquesta comanda té els següents camps:

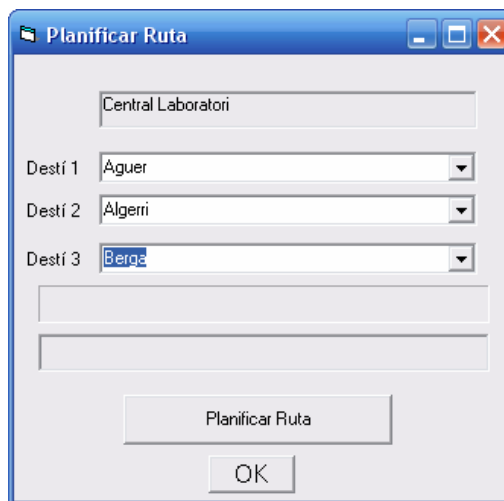
- ✓ Origen: Etiqueta que mostra des d'on es comença la planificació de la ruta. Sempre es començarà des de la central del laboratori.
- ✓ Destí 1,2,3: l'usuari pot seleccionar entre tots els destins possibles. L'ordre dels destins vindrà marcat per com l'usuari seleccioni aquests.
- ✓ Barra de progrés: aquesta barra informa de quin és l'estat de la recerca.
- ✓ Botó "Planificar Ruta": activa l'algoritme de recerca.
- ✓ Etiqueta informativa: etiqueta que mostra missatges a l'usuari informant de quin estat té.



**Figura 32** Pantalla de la comanda “Planificació de rutes”

El funcionament d’aquesta comanda és el següent:

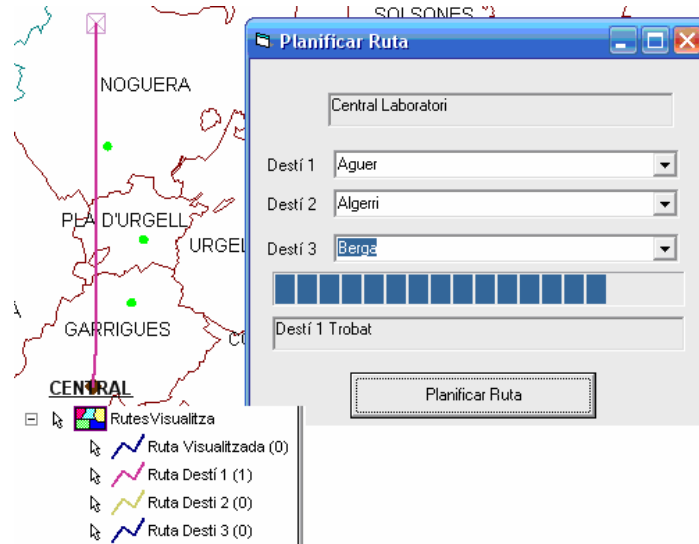
1. L’usuari introdueix quins són els destins que vol planificar en la ruta. En la figura 33 es fa un exemple amb la recerca d’un camí entre els destins: Aguer,Alguerri i Berga.



**Figura 33** Selecció de destins en la comanda “Planificació de rutes”

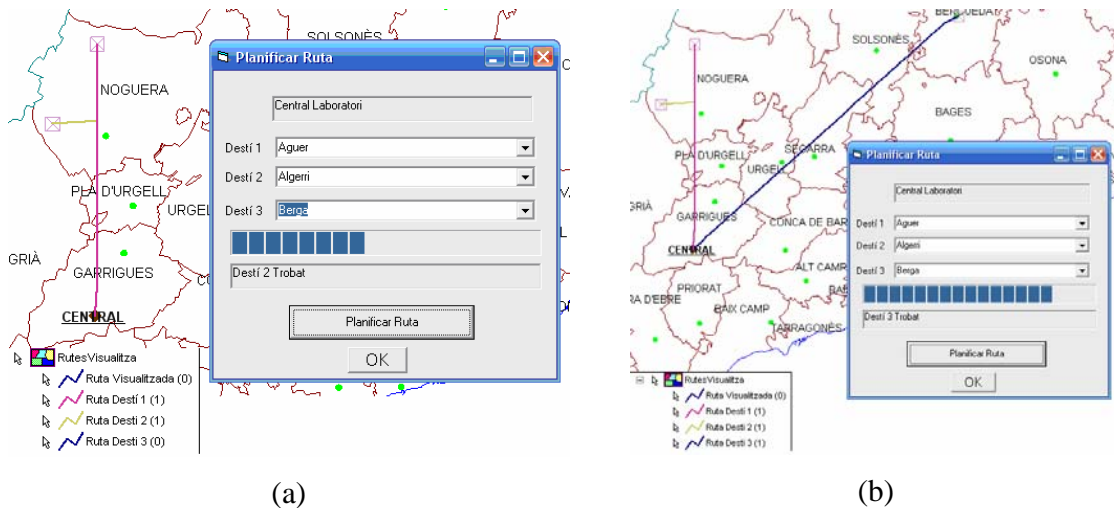
2. L’usuari pressiona el botó “Planificar Ruta”.
3. El sistema busca una ruta entre l’origen i el destí 1. A la figura 34 es mostra el camí que troba. Es pot observar que la llegenda es modifica indicant que la ruta que s’ha

trobat és del destí 1.



**Figura 34** Primer destí trobat.

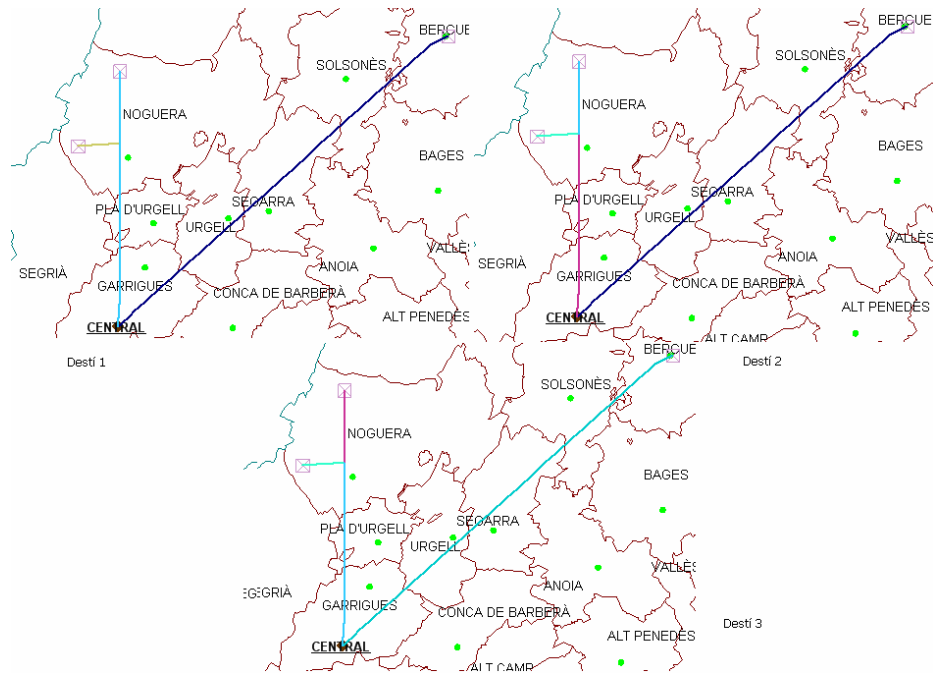
- En el cas d'existir un destí 2 o 3 , el sistema buscarà una ruta entre el destí 1 i el destí 2 o destí 2 i destí 3 respectivament. En el exemple, troba el destí 2 marcat amb una línia groga (veure imatge (a) de la figura 35) i el destí 3 amb una línia blava (veure imatge (b) de la figura 35). De la mateixa manera modifica la llegenda.



**Figura 35** Segon i tercer destins trobats

- El sistema representa en el plànol la ruta resultant, diferenciant per colors les

diferents parts de la ruta (veure imatge (b) de la figura 35). Com que en el exemple algunes rutes es solapen entre elles, en la figura 36 es pot observar quina es la ruta que ha trobat per cada destí marcada en blau turquesa.



**Figura 36** Descripció de les rutes trobades per cada destí

- En el cas de que l'usuari vulgui planificar una nova ruta amb menys destins, tindrà que seleccionar "Seleccionar Destí" en els ComboBox que ja no vulgui utilitzar.

A continuació es mostra el codi font més significatiu.

### 7.2.2.6.3. Codi

---

```

Option Explicit
Dim objConn As Connection
Dim PLine As PolylineGeometry
Dim Ox As Double, Oy As Double, Oz As Double
Dim D1x As Double, D1y As Double, D1z As Double
Dim Mx As Double, My As Double, Mz As Double
Dim D2x As Double, D2y As Double, D2z As Double
Dim D3x As Double, D3y As Double, D3z As Double
Private Sub Form_Load()
    Set objConn = gobjGeoApp.Document.Connections(2)
    Set PLine = gobjGeoApp.CreateService("GeoMedia.PolylineGeometry")
    'Es carga la etiqueta de l'origen.
    Call Cargar_Central
    'Es carga els combobox de la pantalla amb tots els llocs possibles

```

---

```
Call Cargar_Destino1  
Call Cargar_Destino2  
Call Cargar_Destino3
```

'Inicialitza la barra de progrés

```
Barra.Value = 0  
Barra.Min = 0  
Barra.Max = 100
```

**End Sub**

#### **Private Function Cargar\_Destino1()**

'Aquesta funció carrega els combobox, les funcions cargar\_destino2 i cargar\_destino3 són iguals.

```
Dim objRS_Desti1 As GRecordset, objOP_Desti1 As OriginatingPipe  
objConn.CreateOriginatingPipe objOP_Desti1  
objOP_Desti1.Table = "LlocsOD"  
objOP_Desti1.Filter = "Descripcio<>"" & Origen.Caption & ""  
Set objRS_Desti1 = objOP_Desti1.OutputRecordset  
Desti1.AddItem ("Seleccionar Desti")  
If objRS_Desti1.RecordCount() <> 0 Then  
    objRS_Desti1.MoveFirst  
    While Not objRS_Desti1.EOF  
        Desti1.AddItem (objRS_Desti1.GFields("descripcio"))  
        objRS_Desti1.MoveNext  
    Wend  
End If
```

**End Function**

#### **Private Function Cargar\_Central()**

'Aquesta funció carrega la etiqueta d'origen i cerca les coordenades emmagatzemades al sistema per la central del laboratori.

```
Dim objRS_Central As GRecordset, objOP_Central As OriginatingPipe  
Dim GSS As Variant  
Set GSS = gobjGeoApp.CreateService("Geomedia.GeometryStorageService")  
Dim PPoint As PointGeometry  
Set PPoint = gobjGeoApp.CreateService("GeoMedia.PointGeometry")  
objConn.CreateOriginatingPipe objOP_Central  
objOP_Central.Table = "Central"  
Set objRS_Central = objOP_Central.OutputRecordset  
objRS_Central.MoveFirst  
GSS.StorageToGeometry objRS_Central.GFields("Geometry").Value, PPoint  
Ox = PPoint.Origin.x  
Oy = PPoint.Origin.y  
Oz = PPoint.Origin.z  
Origen.Caption = objRS_Central.GFields("Nom")
```

**End Function**

#### **Private Function CoordenadasD1()**

'aquesta funció busca les coordenades exactes del destí seleccionat, aquesta funció és repeteix pels altres destins.

```
Dim objRS_Llocs As GRecordset, objOP_Llocs As OriginatingPipe  
objConn.CreateOriginatingPipe objOP_Llocs  
objOP_Llocs.Table = "LlocsOD"
```

---

```

objOP_Llocs.Filter = "Descripcio="" & Desti1.Text & ""
Set objRS_Llocs = objOP_Llocs.OutputRecordset
If objRS_Llocs.RecordCount() <> 0 Then
  D1x = objRS_Llocs.GFields("Cx").Value
  D1y = objRS_Llocs.GFields("Cy").Value
  D1z = objRS_Llocs.GFields("Cz").Value
End If
Dim objRS_L1 As GRecordset, objOP_L1 As OriginatingPipe
objConn.CreateOriginatingPipe objOP_L1
objOP_L1.Table = "LlocsVisualitza"
Set objRS_L1 = objOP_L1.OutputRecordset
If objRS_L1.Updatable Then
  objRS_L1.AddNew
  objRS_L1.GFields("Cx").Value = D1x
  objRS_L1.GFields("Cy").Value = D1y
  objRS_L1.GFields("Cz").Value = D1z
  objRS_L1.GFields("Geometry").Value = objRS_Llocs.GFields("Geometry").Value
  objRS_L1.Update
End If

```

**End Function**

**Private Function InsertarInicio(x As Double, y As Double, z As Double)**

'Aquesta funció afegeix en la llista de nodes possibles l'origen, fent que el node pare sigui ell mateix, i que l'ordre sigui el 1.

```

Dim objRS_NodosPosibles As GRecordset, objOP_NodosPosibles As OriginatingPipe
objConn.CreateOriginatingPipe objOP_NodosPosibles
objOP_NodosPosibles.Table = "NodosPosibles"
Set objRS_NodosPosibles = objOP_NodosPosibles.OutputRecordset
'Inserció del node d'inici
If objRS_NodosPosibles.Updatable Then
  objRS_NodosPosibles.AddNew
  objRS_NodosPosibles.GFields("Cx").Value = x
  objRS_NodosPosibles.GFields("Cy").Value = y
  objRS_NodosPosibles.GFields("Cz").Value = z

```

'Serà el primer en el recorregut per la llista

```

  objRS_NodosPosibles.GFields("Orden").Value = 1
  objRS_NodosPosibles.Update
  objRS_NodosPosibles.MoveLast

```

'Una vegada afegit retorna el Identificador del node i el inserim com a node pare

```

  objRS_NodosPosibles.Edit
  objRS_NodosPosibles.GFields("idpadre").Value = objRS_NodosPosibles.GFields("id1")
  objRS_NodosPosibles.Update
  InsertarInicio = objRS_NodosPosibles.GFields("id1")

```

End If

**End Function**

**Private Function IrASiguiente(Identificador As Double) As Double()**

'Aquesta funció busca el següent node de la llista, aquest node serà el identificat per la variable 'identificador. Es col·locat en una array on es guarda les coordenades i el seu identificador.

```

Dim Virtual(3) As Double
Dim objRS_NodosPosibles As GRecordset, objOP_NodosPosibles As OriginatingPipe

```

```
objConn.CreateOriginatingPipe objOP_NodosPosibles
objOP_NodosPosibles.Table = "NodosPosibles"
objOP_NodosPosibles.Filter = "Orden=" & Identificador
Set objRS_NodosPosibles = objOP_NodosPosibles.OutputRecordset
Virtual(0) = objRS_NodosPosibles.GFields("Cx").Value
Virtual(1) = objRS_NodosPosibles.GFields("Cy").Value
Virtual(2) = objRS_NodosPosibles.GFields("Cz").Value
Virtual(3) = objRS_NodosPosibles.GFields("id1").Value
IrASiguiente = Virtual
```

**End Function**

#### **Private Function BuscarRutas(x As Double, y As Double, z As Double) As Object**

*'Aquesta funció busca totes les rutes on aparegui el node actual.*

```
Dim objRS_N1 As GRecordset, objOP_N1 As OriginatingPipe
objConn.CreateOriginatingPipe objOP_N1
objOP_N1.Table = "NodosRutes"
objOP_N1.Filter = "(Cx=" & Str(Val(Replace(Str(x), ",", "."))) & " and Cy=" &
Str(Val(Replace(Str(y), ",", "."))) & " and Cz=" & Str(Val(Replace(Str(z), ",", "."))) & ")"
Set objRS_N1 = objOP_N1.OutputRecordset
Set BuscarRutas = objRS_N1
```

**End Function**

#### **Private Function BuscarSiguients(Filtro As String) As Object**

*'Aquesta funció busca tots els nodes possibles que es pot anar des del node actual*

```
Dim objRS_NS As GRecordset, objOP_NS As OriginatingPipe
objConn.CreateOriginatingPipe objOP_NS
objOP_NS.Table = "NodosRutes"
objOP_NS.Filter = Filtro
Set objRS_NS = objOP_NS.OutputRecordset
Set BuscarSiguients = objRS_NS
```

**End Function**

#### **Private Function BuscarNodos(Nx As Double, Ny As Double, Nz As Double, Destino As Integer)**

*'Aquesta és la funció principal*

```
Dim R As String 'guarda les rutes possibles
Dim PadreNodo As Double 'Guarda el node pare
Dim BRutas As GRecordset 'guarda el recordset de les rutes possibles
Dim BNodos As GRecordset 'guarda el recordset dels nodes possibles
```

```
Dim objRS_NodosPosibles As GRecordset, objOP_NodosPosibles As OriginatingPipe
Dim ArrayNodos() As Double
Dim EncontradoDestino As Boolean
Dim NOrden As Double 'Guarda l'ordre en que tenim que anar a la llista a buscar el següent
Dim OrdenN As Double 'Guarda el identificador de quin és l'ordre dins de la llista de cada node
'Es comença tot en 1
NOrden = 1
OrdenN = 1
'Inserció del primer i origen
PadreNodo = InsertarInicio(Nx, Ny, Nz)
```

---



```

R = ""
'inicialització de la variable EncontradoDestino a fals.
EncontradoDestino = False
'Connexió a nodes possibles
objConn.CreateOriginatingPipe objOP_NodosPosibles
objOP_NodosPosibles.Table = "NodosPosibles"
Set objRS_NodosPosibles = objOP_NodosPosibles.OutputRecordset
'Bucle principal mentre no es troba el destí es continua recorrent la llista
While Not EncontradoDestino
'Es busca el node que segueix en l'ordre de recerca establert.
  ArrayNodos = IrASiguiente(NOrden)
  Nx = ArrayNodos(0)
  Ny = ArrayNodos(1)
  Nz = ArrayNodos(2)
  PadreNodo = ArrayNodos(3)
  'Es busquen les rutes on està aquest node.
  Set BRutas = BuscarRutas(Nx, Ny, Nz)

  If BRutas.RecordCount() <> 0 Then
    BRutas.MoveFirst
    While Not BRutas.EOF
'Escollim tant els nodes de davant com els de darrera del node actual i que estiguin en les rutes
'abans trobades.
      R = R + "(idruta=" & BRutas.GFields("idruta").Value & " and (orden=" &
BRutas.GFields("orden").Value + 1 & " or orden=" & BRutas.GFields("orden").Value - 1 & ")) OR "
      BRutas.MoveNext
    Wend
    R = Left(R, Len(R) - 3)
    Set BNodos = BuscarSiguients(R)
    R = ""

'Una vegada trobats els nodes possibles es mira primer si un d'aquest nodes és el destí, en el cas
'de que no sigui el destí, comprovant que els nodes no existeixin en la llista de nodes possibles, en
'el cas de que no existeixi s'afegeix.
    If BNodos.RecordCount() <> 0 Then
      While (Not BNodos.EOF)
        If Not EncontradoDestino Then
          If objRS_NodosPosibles.Updatable Then
            If Not DestinoOK(Destino, BNodos.GFields("cx"), BNodos.GFields("cy"), BNodos.GFields("cz")) Then
              If Noexiste(BNodos.GFields("cx"), BNodos.GFields("cy"), BNodos.GFields("cz")) Then
                objRS_NodosPosibles.AddNew
' Cada vegada que inserim un node nou s'incrementa la variable OrdenN
                OrdenN = OrdenN + 1
                objRS_NodosPosibles.GFields("Cx").Value = BNodos.GFields("cx")
                objRS_NodosPosibles.GFields("Cy").Value = BNodos.GFields("cy")
                objRS_NodosPosibles.GFields("Cz").Value = BNodos.GFields("cz")
                objRS_NodosPosibles.GFields("idpadre").Value = PadreNodo
                objRS_NodosPosibles.GFields("Orden").Value = OrdenN
                objRS_NodosPosibles.Update
              End If
            Else
' En cas de trobar el destí es posa la variable EncontradoDestino a verdader i el guarda en la llista

```

de nodes possibles com últim node.

```

    EncontradoDestino = True
    Pasos.Caption = "Destí Trobat"
    objRS_NodosPosibles.AddNew
        objRS_NodosPosibles.GFields("Cx").Value = BNodos.GFields("cx")
        objRS_NodosPosibles.GFields("Cy").Value = BNodos.GFields("cy")
        objRS_NodosPosibles.GFields("Cz").Value = BNodos.GFields("cz")
        objRS_NodosPosibles.GFields("idpadre").Value = PadreNodo
        objRS_NodosPosibles.GFields("Orden").Value = OrdenN
    objRS_NodosPosibles.Update
    objRS_NodosPosibles.MoveLast
  End If
End If
End If
  End If
  BNodos.MoveNext
Wend
End If
End If
  NOrden = NOrden + 1
'Incrementa en un NOrden per anar a buscar el següent.
  Set BRutas = Nothing
  Set BNodos = Nothing
  Call ActualizarBarra 'Actualitza la barra de progrés.
Wend
'Aquesta funció recorrerà la llista de nodes cap a enrere per trobar el camí resultant.
  Call CrearRuta(objRS_NodosPosibles.GFields("id1").Value, Destino)
  Set objRS_NodosPosibles = Nothing
  Call BorrarListaNodos

```

**End Function**

**Private Function BuscarPunto(indice As Double) As Double()**

'Aquesta funció retorna les coordenades d'un punt en particular de la llista de Nodes

```

  Dim Virtual(4) As Double
  Dim objRS_NodosPosibles As GRecordset, objOP_NodosPosibles As OriginatingPipe
  objConn.CreateOriginatingPipe objOP_NodosPosibles
  objOP_NodosPosibles.Table = "NodosPosibles"
  objOP_NodosPosibles.Filter = "id1=" & indice
  Set objRS_NodosPosibles = objOP_NodosPosibles.OutputRecordset
  Virtual(0) = objRS_NodosPosibles.GFields("Cx").Value
  Virtual(1) = objRS_NodosPosibles.GFields("Cy").Value
  Virtual(2) = objRS_NodosPosibles.GFields("Cz").Value
  Virtual(3) = objRS_NodosPosibles.GFields("id1").Value
  Virtual(4) = objRS_NodosPosibles.GFields("idpadre").Value
  BuscarPunto = Virtual

```

**End Function**

**Private Function CrearRuta(ultimo As Double , d As Integer)**

'Aquesta funció busca els nodes pares de cada node successivament fins que no troba el node que 'el identificador de node i el node pare són iguals que indica que és l'origen.

```

  Dim ArrayNodos() As Double
  Dim PLine As PolylineGeometry

```

```
Set PLine = gobjGeoApp.CreateService("GeoMedia.PolylineGeometry")
Dim TPoint As Point
Set TPoint = gobjGeoApp.CreateService("GeoMedia.Point")
ArrayNodos = BuscarPunto(ultimo)
While ArrayNodos(4) <> ArrayNodos(3)
    TPoint.x = ArrayNodos(0)
    TPoint.y = ArrayNodos(1)
    TPoint.z = ArrayNodos(2)
'El punt trobat és afegit a la línia final
    PLine.Points.Add TPoint
    ArrayNodos = BuscarPunto(ArrayNodos(4))
Wend
    TPoint.x = ArrayNodos(0)
    TPoint.y = ArrayNodos(1)
    TPoint.z = ArrayNodos(2)
' s'afegeix el node inicial
    PLine.Points.Add TPoint
    Set TPoint = Nothing
'Visualitza la ruta
    Call Crear_Visualitzar(PLine,d)
End Function
```

**Private Function DestinoOK(des As Integer, x As Double, y As Double, z As Double)**

'Funció que indica que ha arribat al destí.

```
If des = 1 Then
    If x = D1x And y = D1y And z = D1z Then
        DestinoOK = True
    Else
        DestinoOK = False
    End If
End If
If des = 2 Then
    If x = D2x And y = D2y And z = D2z Then
        DestinoOK = True
    Else
        DestinoOK = False
    End If
End If
If des = 3 Then
    If x = D3x And y = D3y And z = D3z Then
        DestinoOK = True
    Else
        DestinoOK = False
    End If
End If
```

**End Function****Private Function Noexiste(ex As Double, ey As Double, ez As Double)**

'Funció que indica que un node ja existeix en la llista de nodes possibles

```
Dim objRS_ex As GRecordset, objOP_ex As OriginatingPipe
objConn.CreateOriginatingPipe objOP_ex
```

---

```

    objOP_ex.Table = "NodosPosibles"
    objOP_ex.Filter = "Cx=" & Str(Val(Replace(Str(ex), ",", "."))) & " and Cy=" &
    Str(Val(Replace(Str(ey), ",", "."))) & " and Cz=" & Str(Val(Replace(Str(ez), ",", ".")))
    Set objRS_ex = objOP_ex.OutputRecordset
    If objRS_ex.RecordCount() = 0 Then
        Noexiste = True
    Else
        Noexiste = False
    End If

```

**End Function**

**Private Sub Planificar\_Click()**

'Acció del botó "planifica ruta"

Call BorrarListaNodos

Call BorrarRutasVisualitza

Call Cargar\_Central

If Desti1.Text <> "Seleccionar Desti" Then

Pasos.Caption = "Cercant Ruta 1er Destí"

Barra.Value = 0

Call CoordenadasD1 'Es busca les coordenades del destí 1

Call BuscarNodos(Ox, Oy, Oz, 1) 'Es comença la recerca del camí començant des de la central

If Desti2.Text <> "Seleccionar Destí" Then

Barra.Value = 0

Call CoordenadasD2

Call BuscarNodos(D1x, D1y, D1z, 2)

If Desti3.Text <> "Seleccionar Destí" Then

Barra.Value = 0

Call CoordenadasD3

Call BuscarNodos(D2x, D2y, D2z, 3)

End If

End If

Else

MsgBox ("Ha de seleccionar mínim un destí")

End If

**End Sub**

**Private Function Crear\_Visualitzar(Linea As PolylineGeometry, d As Integer)**

Dim objRS\_RutesV As GRecordset, objOP\_RutesV As OriginatingPipe

objConn.CreateOriginatingPipe objOP\_RutesV

objOP\_RutesV.Table = "RutesVisualitza"

Dim GSS As Variant

Set objRS\_RutesV = objOP\_RutesV.OutputRecordset

Dim GeomOut As Variant

Set GSS = gobjGeoApp.CreateService("Geomedia.GeometryStorageService")

If objRS\_RutesV.Updatable Then

objRS\_RutesV.AddNew

GSS.GeometryToStorage Linea, GeomOut

objRS\_RutesV.GFields("Geometry").Value = GeomOut

objRS\_RutesV.GFields("OrigenRuta").Value = d 'Indica a quin destí pertany la ruta

objRS\_RutesV.Update

End If

---

Call Refrescar

**End Function**

---

Una vegada explicat el projecte tant en la part pràctica com en la teòrica es mostren les conclusions del projecte, així com les noves línies de treball que es poden fer.

---

---

## 8. CONCLUSIONS

---

---

Una vegada finalitzat aquest projecte es poden treure les següents conclusions:

La primera és analitzar que els objectius s'han complert, per això s'exposarà cada objectiu que es va plantejar i quin ha estat el resultat:

- ✓ Conèixer les característiques principals d'un SIG, quins components ho formen ,així com quines són les aplicacions que tenen.

Aquest objectiu s'ha portat a terme amb èxit, per una banda s'ha après que és un SIG i quines són les seves característiques, també s'ha après els conceptes principals de la cartografia i la geodèsia, conceptes que han ajudat d'una manera molt important a poder resoldre l'objectiu de crear un SIG personalitzat.

- ✓ Conèixer el programari Geomedia Professional 6.0.

Aquest objectiu també ha estat resolt amb èxit. Geomedia Professional 6.0 és un programari del que no es tenia cap mena d'informació i ha estat molt interessant aprofundir en aquest programari i poder veure totes les possibilitats que té.

- ✓ Conèixer alguns tipus d'algoritmes d'optimització de rutes, per tal de resoldre les funcionalitats que planteja l'enunciat.

Aquesta part era la part més nova del projecte, i era el repte ha aconseguir. S'ha investigat en el món dels grafs i s'ha pogut observar com és una ciència amb moltes possibilitats per poder treure profit. En aquest cas s'ha investigat en la recerca de camins i s'ha pogut veure com es pot arribar a crear uns algoritmes veritablement eficients i que poden proporcionar molta ajuda a l'hora de fer recerca de rutes i camins.

- ✓ Construir un sistema SIG i programar comandes amb Geomedia Professional 6.0 i Visual Basic 6.0.

Aquesta part ha estat la part més tècnica del projecte, s'ha treballat sobre Visual Basic 6.0 amb ajuda del *wizard command* de Geomedia Professional 6.0 i s'ha arribat a programar el sistema. També s'ha pogut conèixer totes les possibilitats que dona l'ICC, el qual ha estat una eina molt important en la construcció del sistema.

En general s'ha pogut observar com un SIG es pot acostar d'una manera molt acurada a un problema real, i com aquesta tecnologia pot ajudar a resoldre problemes.

Després de fer les conclusions d'aquest projecte es poden plantejar noves funcionalitats o millores del sistema.

### **8.1. Noves línies de treball**

Aquest projecte pot tenir les següents línies futures de treball:

- ✓ Optimització dels camins: intentar optimitzar els algorismes de recerca de camins per poder trobar els camins més curt o el més ràpid etc..
- ✓ Ampliar informació camins: poder emmagatzemar velocitats màximes o mínimes , així com punts de parades, direccions, peatges, etc...
- ✓ Ampliar informació destins: poder emmagatzemar informació extra dels destins, tipus d'animals, nom propietari, etc..
- ✓ Utilització d'una BD Oracle o Microsoft Sql Server, per tal de guanyar en seguretat i escalabilitat.

---

---

## 9. BIBLIOGRAFIA

---

---

- [1] Departament d'expressió gràfica, disseny i projectes de la Universitat de Màlaga. <http://exp-grafica.uma.es/Profesores/www-jrad/document/gis/sig.pdf>, presentació en Microsoft PowerPoint sobre sistemes d'informació geogràfica.(Setembre 2006)
- [2] Bosque Sendra,Joquin. "Sistemas de información geográfica", 2ª edición, ISBN:84-321-3154-7, Editorial RIALP.
- [3] Comas,David; Ruiz,Ernest. "Fundamentos de los sistemas de información geográfica", ISBN: 84-344-3452-0
- [4] Memòria projecte "Implantació d'un SIG amb Geomedia Professional 5.2 i l'enllaç amb Epanet", Martí Pàmies Solà, UOC.
- [5] Memòria projecte,"Construcción de un SIG municipal para la gestión de una agencia de taxis", Raúl Bautista Eructes,UOC
- [6] Wikipedia. Definició de cartografia [<http://es.wikipedia.org/wiki/Cartograf%C3%ADa> ,(Octubre 2006)]
- [7] Wikipedia.Definició de Meridià [<http://es.wikipedia.org/wiki/Meridiano> ,(Octubre 2006)]
- [8] Wikipedia.Definició de Paral·lel [<http://es.wikipedia.org/wiki/Paralelo> ,(Octubre 2006)]
- [9] Escola Nàutica de Catalunya. Imatge de representació dels meridians i els paral·lels [[www.enc.es/aulavirtual/0\\_visita\\_PER/PER\\_CD/Contenidos/imag\\_intera/navegacion/01lat\\_lon/lat\\_lo ng3.html](http://www.enc.es/aulavirtual/0_visita_PER/PER_CD/Contenidos/imag_intera/navegacion/01lat_lon/lat_lo ng3.html)], (Octubre 2006)]
- [10] Cartesia.org Geodèsia  
[[www.cartesia.org/data/apuntes/nociones/Nociones\\_de\\_Geodesia\\_y\\_GPS.pdf](http://www.cartesia.org/data/apuntes/nociones/Nociones_de_Geodesia_y_GPS.pdf) ], (Octubre 2006)]
- [11] Cartesia.org Cartografia  
[[www.cartesia.org/data/apuntes/nociones/Nociones\\_de\\_Cartografia.pdf](http://www.cartesia.org/data/apuntes/nociones/Nociones_de_Cartografia.pdf)], (Octubre 2006)]
- [12] Wikipedia. Geoide [<http://es.wikipedia.org/wiki/Geoide> ,(Octubre 2006)]
- [13] Xarxa telemàtica educativa de Catalunya. Projeccions  
[[www.xtec.es/~mroman/Projecte/projecciones.htm](http://www.xtec.es/~mroman/Projecte/projecciones.htm) ], (Octubre 2006)]
- GRAFOS
- [14] Wikipedia. Grafts [<http://es.wikipedia.org/wiki/Grafos> ,(Octubre 2006)]
- [15] Universitat Politècnica de Madrid. Teoria dels GraftsPagina  
[[www.dma.fi.upm.es/docencia/segundociclo/teorgraf](http://www.dma.fi.upm.es/docencia/segundociclo/teorgraf) ], (Octubre 2006)]
- [16] Wikipedia. Coordenades UTM [[http://es.wikipedia.org/wiki/Coordenadas\\_UTM](http://es.wikipedia.org/wiki/Coordenadas_UTM) ], (Octubre 2006)]



[17] Portal de geografia. [[http://podespacial.com/index.php?post\\_category=Geodesia](http://podespacial.com/index.php?post_category=Geodesia), (Gener 2007)]

[18] [Intergraph](#). Manual de l'usuari.

[19] "Instituto Internacional en Conservación y Manejo de Vida Silvestre" (ICOMVIS) .Proyecciones [[www.icomvis.una.ac.cr/telesig/pdf/proyeccion\\_datum\\_teoría.pdf](http://www.icomvis.una.ac.cr/telesig/pdf/proyeccion_datum_teoría.pdf) , (Octubre 2006)]

[20] Pàgina sobre les coordenades UTM de la Universitat Autònoma de Madrid [http://www.uam.es/personal\\_pdi/ciencias/casado/GEORED/Topo-1/coordenadas\\_utm.htm](http://www.uam.es/personal_pdi/ciencias/casado/GEORED/Topo-1/coordenadas_utm.htm) , (Octubre 2006)

[21] Institut cartogràfic de Catalunya <http://www.icc.es> , (Octubre 2006)

[22] Manual d'aprenentatge de Geomedia Professional 6.0

[23] Manual de l'usuari de Geomedia Professional 6.0

[24] Pàgina sobre projeccions UTM de d'institut geogràfic Militar de Chile [http://www.igm.cl/Proyeccion\\_utm.html](http://www.igm.cl/Proyeccion_utm.html) , (Novembre 2006)

[25] Pàgina sobre l'algoritme A\* escrit per Diego G. Ruiz del portal de programació dedalus-software <http://www.dedalus-software.com.ar/tutoriales/pathfinding/astar.pdf> (Novembre 2006)

[26] Pàgina sobre l'algoritme ANCHO-PRIMERO Profundidad-primero escrit per Diego G. Ruiz del portal de programació dedalus-software <http://www.dedalus-software.com.ar/tutoriales/pathfinding/pathfinding.pdf>

(Novembre 2006)

[27] Manual del desenvolupador de Geomedia Professional 6.0

[28] Árboles, Sergio i Navarro, Luis. "Visual Basic 6. Curso de iniciación", ISBN:84-89700-96-6, Editorial INFORBOOK'S, S.L.