

1509

ENSSIB

Ecole Nationale Supérieure des Sciences de
l'Information et des Bibliothèques

**Université
Claude Bernard - Lyon 1**

DESS en INFORMATIQUE DOCUMENTAIRE

Rapport de Stage

INTERNET et LES CARTES GEOGRAPHIQUES

Création d'un répertoire thématique sur le Web en vue d'enrichir le site existant
Recherche et réflexion sur la réalisation d'une interface cartographique dynamique

MILLAN LOPEZ María Teresa

Sous la direction de
Philippe COCHEREI
ACTUALIS DATA
109, cours Emile Zola
69100 Villeurbanne

1999



ENSSIB

Ecole Nationale Supérieure des Sciences de
l'Information et des Bibliothèques

**Université
Claude Bernard - Lyon 1**

DESS en INFORMATIQUE DOCUMENTAIRE

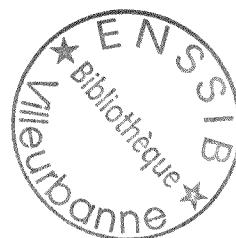
Rapport de Stage

INTERNET et LES CARTES GEOGRAPHIQUES

Création d'un répertoire thématique sur le Web en vue d'enrichir le site existant
Recherche et réflexion sur la réalisation d'une interface cartographique dynamique

MILLAN LOPEZ María Teresa

Sous la direction de
Philippe COCHEREL
ACTUALIS DATA
109, cours Emile Zola
69100 Villeurbanne



1999

1999
11) ST
14

REMERCIEMENTS

*Merci mon Dieu !, tu qui est toujours avec moi
et ne me laisses pas. Je ne peux pas oublier ton aide,
ta présence dans ma vie et ton amour tendre,
sans toi, rien ne serait possible.
Je t'aime de tout mon cœur !*

Je remercie M. Lardy pour son aide, son conseil et pour la relecture de mon mémoire.

Je remercie tout spécialement M. Philippe Cocherel de m'avoir accueilli au sein de l'entreprise, pour sa confiance, sa disponibilité, son soutien, son aide au cours du stage et aussi pour la relecture de mon mémoire.

Je tiens à remercier également Xavier, Catherine et Izumi pour leur aide, leur attention, leur disponibilité et leur bonne humeur qui ont permis d'avoir un bon environnement de travail.

Je remercie de tout mon cœur Joëlle Razafinjato pour son amitié et son soutien tout au long de cette année.

Je n'oublie pas José André Francisco pour son aide, ses encouragements et sa grande amitié. Je tiens également à remercier tous mes amis qui m'appuient de loin et me font confiance.

Et bien sûr, à Lucy qui m'a soutenue tout le temps, pour sa patience, son amitié, sa compagnie et son amour inconditionnel.

Enfin, je ne peux pas oublier toute ma famille qui me soutient, je la remercie pour son amour, sa confiance et ses prières. Mamá, Papá, hermanos : « Je vous aime beaucoup ! »

Titre : Internet et les cartes géographiques. Création d'un répertoire thématique sur le Web en vue d'enrichir le site existant. Recherche et réflexion sur la réalisation d'une interface cartographique dynamique.

Auteur : María Teresa MILLAN LOPEZ

Résumé

Dans ce rapport, après une présentation du contexte de travail, nous exposerons tout d'abord les modifications apportées au site existant et dans un second temps, les démarches nécessaires à la réalisation d'un répertoire thématique sur Internet. Enfin, nous présenterons les techniques et résultats issus d'une recherche pour créer une interface Web avec une carte géographique vectorielle.

Mot clés : Internet, Site, Site Web, Répertoire, Indexation, Thesaurus, Représentation cartographique, Cartographie, Géographie, Carte, Carte géographique, Réseau WWW, Image, Imagerie, coordonnées géographiques, Tourisme, Loisir.

Title : Internet and the geographic maps. Creation of a thematic Web repertory in order to enrich the existing site. Research on the realisation of a vectorial cartographic interface.

Abstract

In this report, after the presentation of the work context, we will present the modifications achieved on the existing site. Secondly, we will develop the approach for realising a thematic directory on the Internet. Finally, we will present the techniques and results of a research on the creation of a interface with a vectorial geographic map on the Web.

Keywords : Internet, Site, Web Site, Directory, Indexing, Thesaurus, Cartographic Representation, Cartography, Geography, Maps, Map (geography), World wide web, Image, Imagery, geographic coordinates, Tourisme, Recreation.

Table des matières

INTRODUCTION.....	2
1. PRÉSENTATION DE LA SOCIÉTÉ.....	3
1.1. MISSIONS	3
1.2. FONDS DOCUMENTAIRE	4
1.3. ENVIRONNEMENT DE TRAVAIL.....	4
1.3.1. <i>Espace de travail</i>	4
1.3.2. <i>Logiciels les plus utilisés au cours du stage</i>	5
2. PRÉSENTATION DU SITE EXISTANT	6
2.1. PRÉSENTATION DE L'INTERFACE WEB.....	6
2.2. PRÉSENTATION DE LA BASE DE DONNÉES.....	8
3. PRÉSENTATION DES MISSIONS ASSIGNÉES AU COURS DU STAGE.....	10
3.1. MODIFICATIONS DU SITE EXISTANT.....	10
3.1.1. <i>Recherche par ville</i>	12
3.1.2. <i>Calendrier</i>	14
3.1.3. <i>Une seule page réponse</i>	15
3.2. RÉPERTOIRE THÉMATIQUE DES SITES DE VILLES.....	16
3.2.1. <i>Recherche des villes touristiques de France</i>	17
3.2.2. <i>Recherche des sites Internet des villes</i>	17
3.2.3. <i>Indexation des villes</i>	20
3.2.4. <i>Thesaurus touristique et des loisirs</i>	20
3.2.5. <i>Réalisation d'une Interface des villes</i>	21
3.2.5.1. Modification de la Base de Données existante.....	22
3.2.5.2. Conception de la rubrique des villes.....	31
3.3. APPROCHE CARTOGRAPHIQUE.....	36
3.3.1. <i>Quelques Solutions « Statiques »</i>	37
3.3.1.1. HTML.....	37
3.3.1.2. Image réactive.....	38
3.3.1.3. HTML Dynamique, Active X.....	39
3.3.2. <i>Solutions dynamiques</i>	41
3.3.2.1. MapInfo et NetMaps.....	41
3.3.2.2. Les Servlets.....	44
CONCLUSION.....	52
ANNEXES.....	54
BIBLIOGRAPHIE.....	101

Introduction

Internet est devenu un outil incontournable et peut même être l'objet de l'activité principale de l'entreprise. En effet, il est à la fois une source d'information gigantesque et un moyen de communication, de diffusion, d'acquisition, d'apprentissage et de loisir sans précédent. Le rôle de l'entreprise ACTUALIS DATA est de proposer au public un répertoire thématique offrant une information sur plus de 3 500 festivals en France.

En vue de rendre plus attractif, complet et compétitif le site existant, les responsables de l'entreprise ont décidé d'apporter de nouvelles fonctionnalités:

- Pour le rendre plus convivial,
 - Modifications de la rubrique Actualité des festivals
- Pour l'enrichir,
 - Répertoire francophone
 - Attractions touristiques et culturelles
 - Approche cartographique

Les tâches qui m'ont été confiées furent : les modifications de la rubrique Actualité des festivals, la création d'un répertoire d'adresses Internet des villes touristiques de France ainsi que la recherche et la réflexion concernant une interface avec carte géographique.

D'abord, une présentation de l'entreprise et de l'environnement du travail s'avère nécessaire pour comprendre le fonctionnement, la logique, les techniques utilisées et les attentes de l'entreprise. Dans un deuxième temps, nous présenterons les différentes missions réalisées au sein de l'entreprise et les démarches effectuées.

1. Présentation de la société

1.1. Missions

ACTUALIS DATA est une micro-entreprise créée en 1997, spécialisée dans les informations culturelles. Elle enrichit depuis deux ans une base de données portant sur plus de 3500 manifestations nationales, festivals, programmations saisonnières et grandes fêtes (commentaires et articles, programmes, indexation thématique, descripteurs, adresses professionnelles et informations pratiques destinées aux festivaliers) dont certaines informations sont consultables gratuitement sur le site Internet <http://www.muser.tm.fr> depuis janvier 1999 (210.000 pages vues). L'entreprise prépare pour fin 1999 plusieurs produits : la vente de sa base de données à plusieurs sites portails en quête de contenus culturels et le développement d'un site Internet fédérateur des festivals en France (le site Muser).

Sur le plan économique, le festival est un outil de développement touristique pour lequel d'importantes sommes d'argent sont investies. De nombreuses régions, départements et villes se sont dotés de cellules culturelles organisatrices d'événement culturels qui permettent de compléter l'offre touristique et augmenter la fréquentation touristique.

La niche économique des festivals n'est pas négligeable : elle représente probablement plusieurs milliards de FF en France. Elle est fortement liée à plusieurs autres secteurs tels que les médias (presse, radio, TV, affichage, Internet), la publicité (graphistes, régies) , la création artistique (musiciens, acteurs...), les prestataires techniques (son, lumière, scène..), les réseaux de billetterie et bien sûr, les institutions publiques.

La présentation d'un guide sur les festivals en France sur Internet présente de nombreux avantages :

- la possibilité de mettre à jour en temps réel les informations reçues de la part des festivals (dates, programmes..) ;
- les médias traditionnels diffusent actuellement de la publicité ainsi que les programmes des festivals (cahiers spéciaux, guides estivaux). Il n'existait pas de structure similaire sur Internet ;
- profiter de la durée limitée de ces événements : obligation des festivals à diffuser massivement de la publicité et leurs programmes quelques semaines avant l'événement. Une partie importante des budgets sont destinés à la publicité / communication.

1.2. Fonds documentaire

La société dispose d'un fonds documentaire couvrant 3 500 événements en France sous forme de dossiers de presse, brochures publicitaires, programmes, sources d'informations issues d'institutions culturelles (Comité Régionaux du Tourisme, Comité Départementaux du Tourisme, Office de Tourisme).

1.3. Environnement de travail

1.3.1. Espace de travail

Pour la mission proposée par la société Actualis DATA, il a été nécessaire de disposer d'une connexion Internet permanente pendant un mois et demi, afin de mener à bien la recherche de sites de villes présentant des particularités touristiques , mais également celle concernant la cartographie.

Par ailleurs, pour développer le site, il s'est avéré nécessaire d'avoir un espace de travail distinct pour faire des tests sur les scripts. Nous avons utilisé un répertoire de tests sous la racine WWW du serveur Windows NT ayant Internet Information Server, accessible par le réseau local Ethernet.

1.3.2. Logiciels les plus utilisés au cours du stage

PowerMark

Logiciel d'indexation des signets « favoris » et de veille Internet. Logiciel de veille Internet. Il nous a permis de stocker localement les URL choisies ainsi que d'autres caractéristiques requises pour leur indexation. De plus, ce logiciel permet de vérifier automatiquement l'existence, le non fonctionnement, ou la suppression des URL stockées.

FrontPage

Ce logiciel a permis de créer un espace de travail personnel. Nous avons créé la réplique exacte du site Muser officiel (répertoires, fichiers, structure, liens etc...). Le site de test était installé sur le serveur Windows NT afin de recréer les conditions d'accès à distance. FrontPage permet aussi de stocker les pages HTML, ASP ainsi que les scripts.

MapInfo

Nouveau logiciel dans l'entreprise permettant la conception de cartes géographiques : mode point (format GIF, JPG etc.) et vectorielles (avec une base de données cartographique). Il a fallu maîtriser son fonctionnement général en vue de son utilisation et de l'exploitation de la cartographie sur le Web.

Access

La base de données existant dans l'entreprise était développée sous Access. Pour effectuer les modifications sur le site, nous avons dû modifier la base de données en prenant en compte les règles de tout système de Gestion de BD.

JDK1.2

Java Development Kit est un logiciel qui permet de programmer avec le langage Java. Il est composé par des classes qui permettent de faire la conception d'applets, de servlets (voir Chapitre 3.3.2.ii) et d'autres types d'application. En plus il contient les programmes pour compiler (javac.exe) et exécuter (java) les programmes écrits en java, entre autres.

JSDK2.1/JSDK2.0

Java Servlet Development Kit est un logiciel qui permet d'avoir un serveur de servlets. Une fois démarré ce serveur, les servlets peuvent s'exécuter sans problème dans leur répertoire prédéfini (voir Chapitre 3.3.2.ii).

2. Présentation du site existant

2.1. Présentation de l'Interface web

Avant de commencer les modifications du site, il fallait connaître le système en place ; son fonctionnement, les techniques et la logique utilisées.

Le site est constitué de 4 rubriques principales:



Festivals par thème



Cette rubrique propose une recherche thématique, comme par exemple : les festivals de musique, de danse etc... Le système d'indexation actuel permet de combiner plusieurs variables : Beaux lieux + Musique Jazz + entre 100 et 200 FF.

Actualité des festivals

Les festivals qui se déroulent dans le mois en cours. La recherche proposée est thématique. De plus, une sous-rubrique propose chaque mois une sélection de plusieurs festivals particulièrement intéressants.

Recherche par mot clé

Cette rubrique propose une recherche basique par mot clé en plein texte sur la base de données.

Rubrique 100% Web

Enfin, cette rubrique recense et permet l'accès à plus de 700 festivals qui possèdent un site Internet.

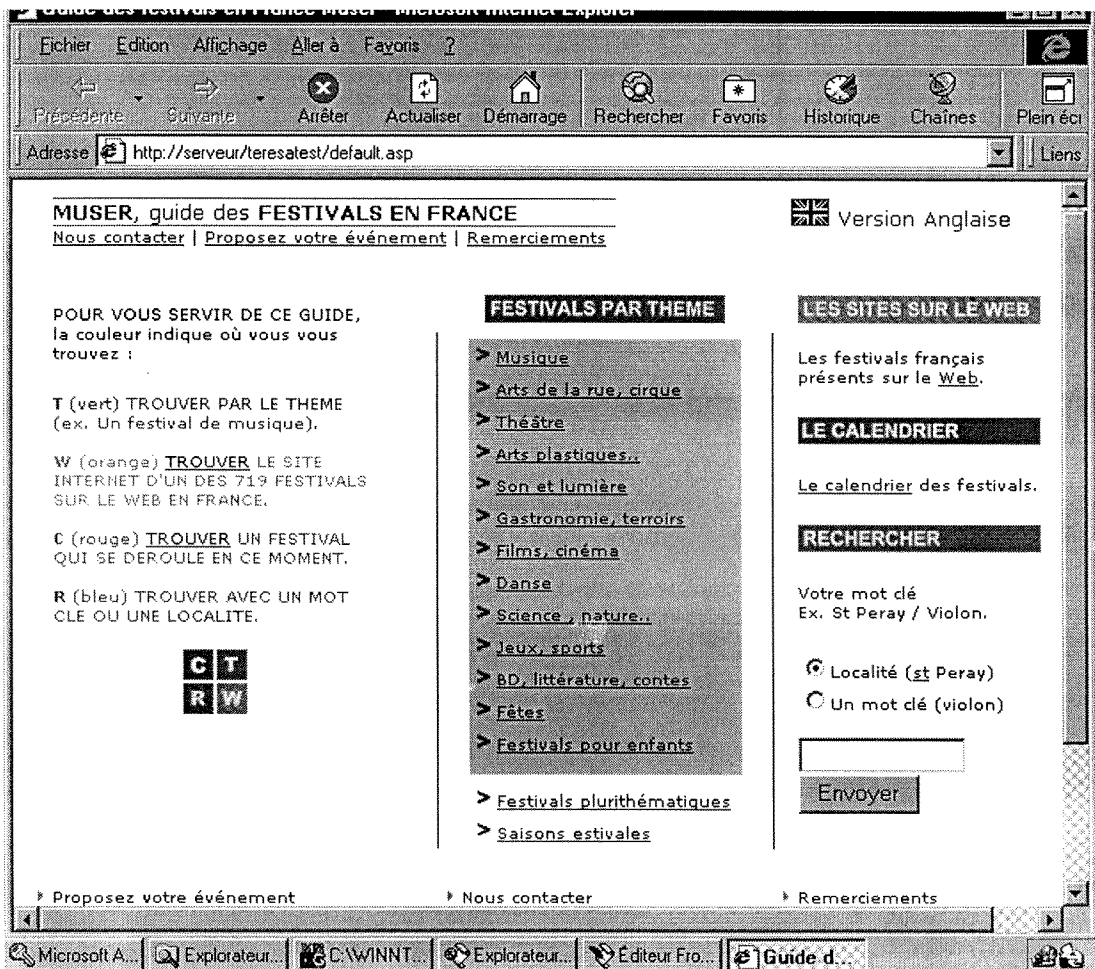


Figure 1. Page d'accueil du site existant

Par ailleurs, les outils utilisés pour la conception du site sont : le langage Vbscript, JavaScript, ActiveX, HTML et les pages ASP (Active Server Pages) (cf. Figure 1). De plus, il y a une Base de Données ACCESS qui est exploitée sur l'Internet à l'aide des pages ASP, dans laquelle sont stockées toutes les données sur les festivals.

2.2. Présentation de la Base de Données

La Base de Données (BD) existante est une BD ACCESS et son Modèle Relationnel de Données avant d'être modifié avait la forme suivante :

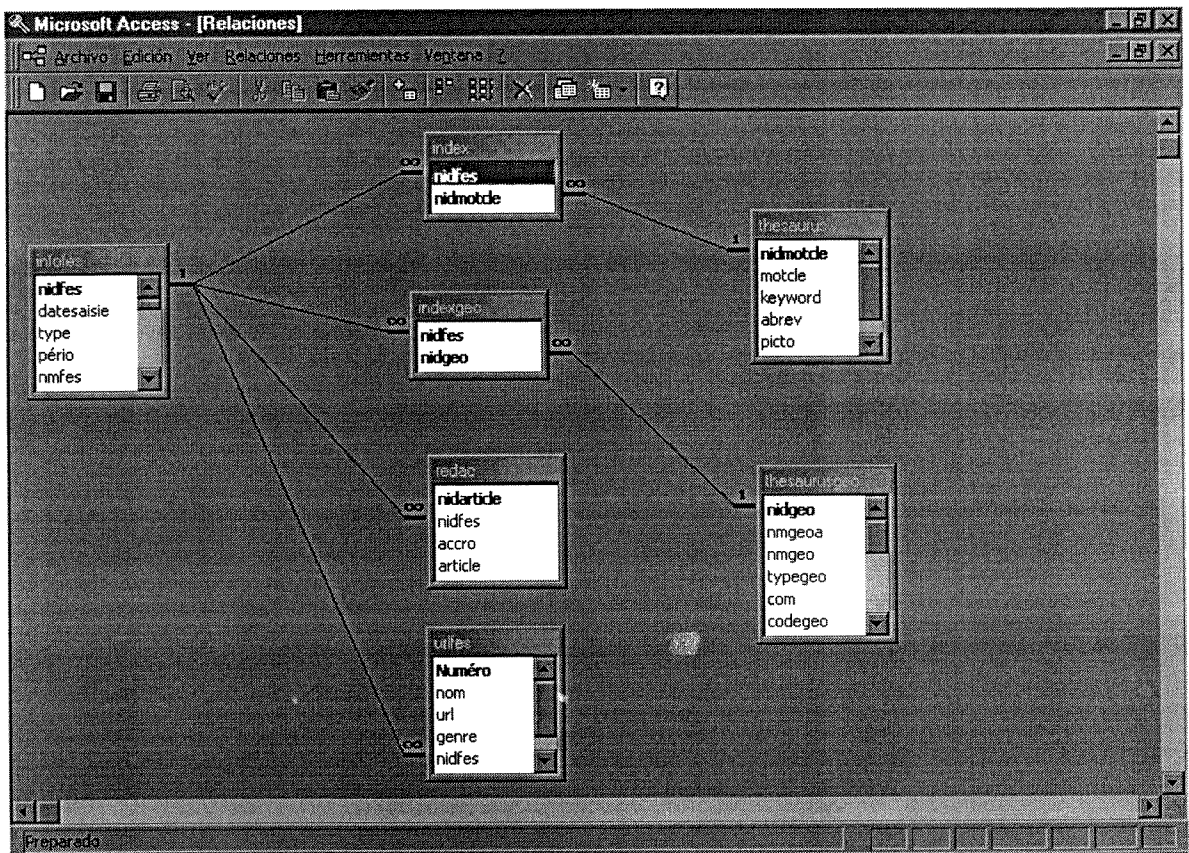


Figure 2. Modèle Relationnel de la Base de Données existante

La BD ACCESS contenait 7 tables dont 2 sont des tables intermédiaires qui ont été utilisées pour défaire les relations N-N dans le Modèle Conceptuel de Données (MCD). Pour les tables restantes, la description est la suivante :

Table	Description	Champs
Infofes	Cette table contient l'information de chaque festival indexé sur le site.	Nom du festival, date de début, date de fin, lieu, périodicité...
Urlfes	Urlfes contient l'adresse ou les adresses URL de chaque festival s'il y en a.	Nom du site, URL, adresse électronique ...
Redac	Celle-ci contient des articles propres à chaque festival, lesquels ont été rédigés par la société.	Article, numéro du festival, accro...
Thesaurus	Cette table contient les termes servant à l'indexation des festivals, comme par ex. : la musique (jazz), la danse, le théâtre, le cinéma etc...	Numéro, mot, abréviation, nom fichier *.gif associé....
Thesaurusgeo	Cette table contient toutes les régions de France, ses départements, ses villes et villages.	Numéro, code postal, nom (région, dept. ville ou village), commentaire...

Relations et cardinalités existantes :

Infofes ¹ _____ ^N redac

Un festival (infofes) peut avoir un ou plusieurs articles (redac), et un article est conçu pour un seul festival.

Infofes ¹ _____ ^N urlfes

Un festival (infofes) peut n'avoir aucune ou plusieurs adresses URL (urlfes), et une adresse URL contient l'information d'un seul festival.

Infofes ^N _____ ^N thesaurus

Un festival (infofes) peut être indexé avec un ou plusieurs mots du thesaurus (thesaurus), et un mot du thesaurus peut servir à indexer un ou plusieurs festivals.

Infofes ^N _____ ^N thesaurusgeo

Un festival (infofes) peut se dérouler dans une ou plusieurs communes (thesaurusgeo), et dans une seule commune se déroulent un ou plusieurs festivals.

3. Présentation des missions assignées au cours du stage.

Cette partie est consacrée aux démarches suivies pour chaque tâche, afin d'aboutir aux objectifs envisagés. En premier lieu, nous présenterons les modifications apportées à la rubrique Actualités de festivals, puis les démarches faites pour la création du répertoire thématique de villes touristiques de France. Enfin, nous présenterons différentes solutions analysées pour l'insertion de l'interface cartographique vectorielle sur le Web.

3.1. Modifications du site existant

L'entreprise avait identifié plusieurs besoins dont les solutions devaient être adaptées à la nouvelle version du site. Il s'agissait notamment d'améliorer la rubrique d'actualité des festivals existante (*cf. Figure 3*). Ce travail consistait à offrir aux visiteurs du site une nouvelle façon d'effectuer leur recherche.

Il avait été convenu de travailler, d'abord, avec l'actualité des festivals de musique (*cf. Figure 4*). Nous avons donc proposé d'introduire :

- non seulement une recherche thématique, mais aussi une recherche par jour, avec un calendrier et une recherche sur toutes les villes où se déroulent les festivals indexés dans le site.

- une seule page ASP réponse (une fois les requêtes posées), pour simplifier la maintenance du site.
- l'utilisation des **variables session**, lesquelles sont stockées sur le serveur pour chaque connexion d'un client avec le site. Cela permet de soulager la base de données (ne pas exécuter plusieurs fois une même requête), de stocker des valeurs dans ces variables et les réutiliser pendant toute la session d'un utilisateur.

D'abord, nous présenterons les actions entreprises afin d'insérer dans la page ASP un calendrier dynamique, puis les actions pour l'insertion d'une liste déroulante ayant les villes de festivals indexées sur la Base de Données. Enfin, nous présenterons la page ASP réponse, exécutée une fois le jour du calendrier choisi ou la ville sélectionnée.

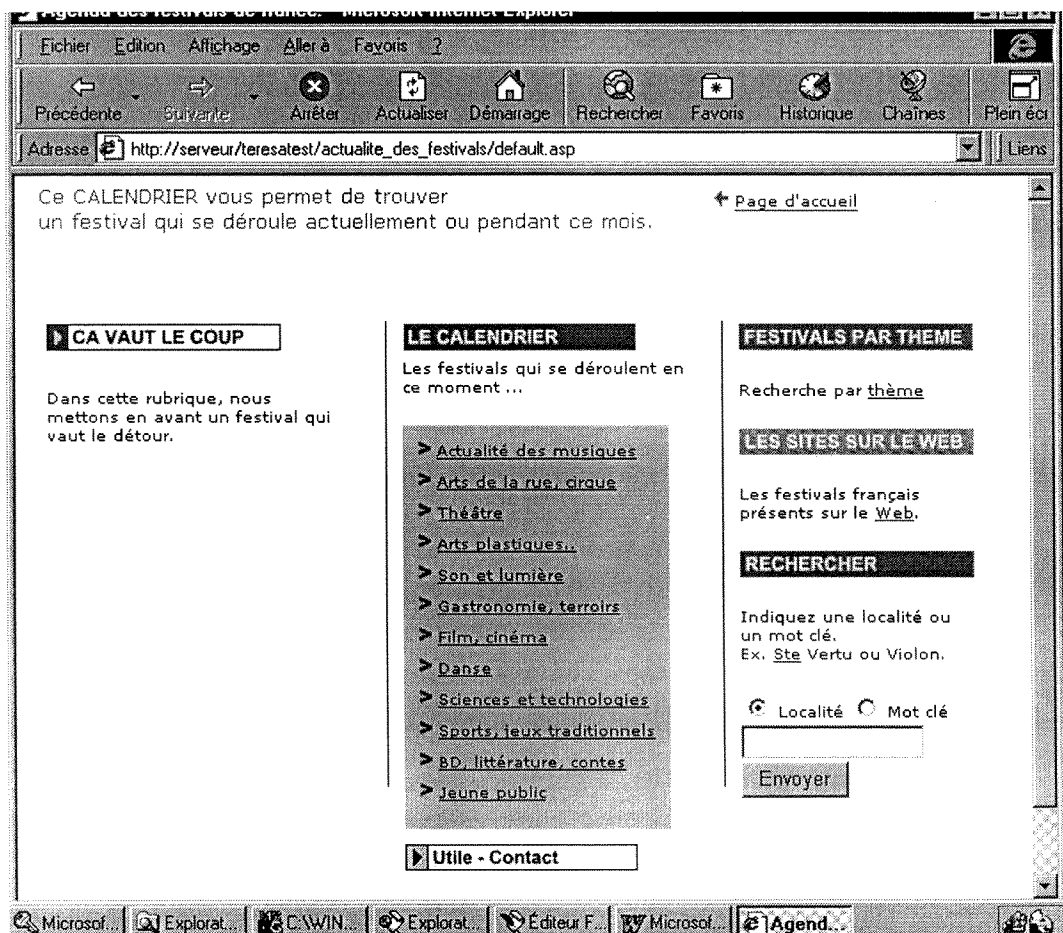


Figure 3. Rubrique Actualité de festivals existant.

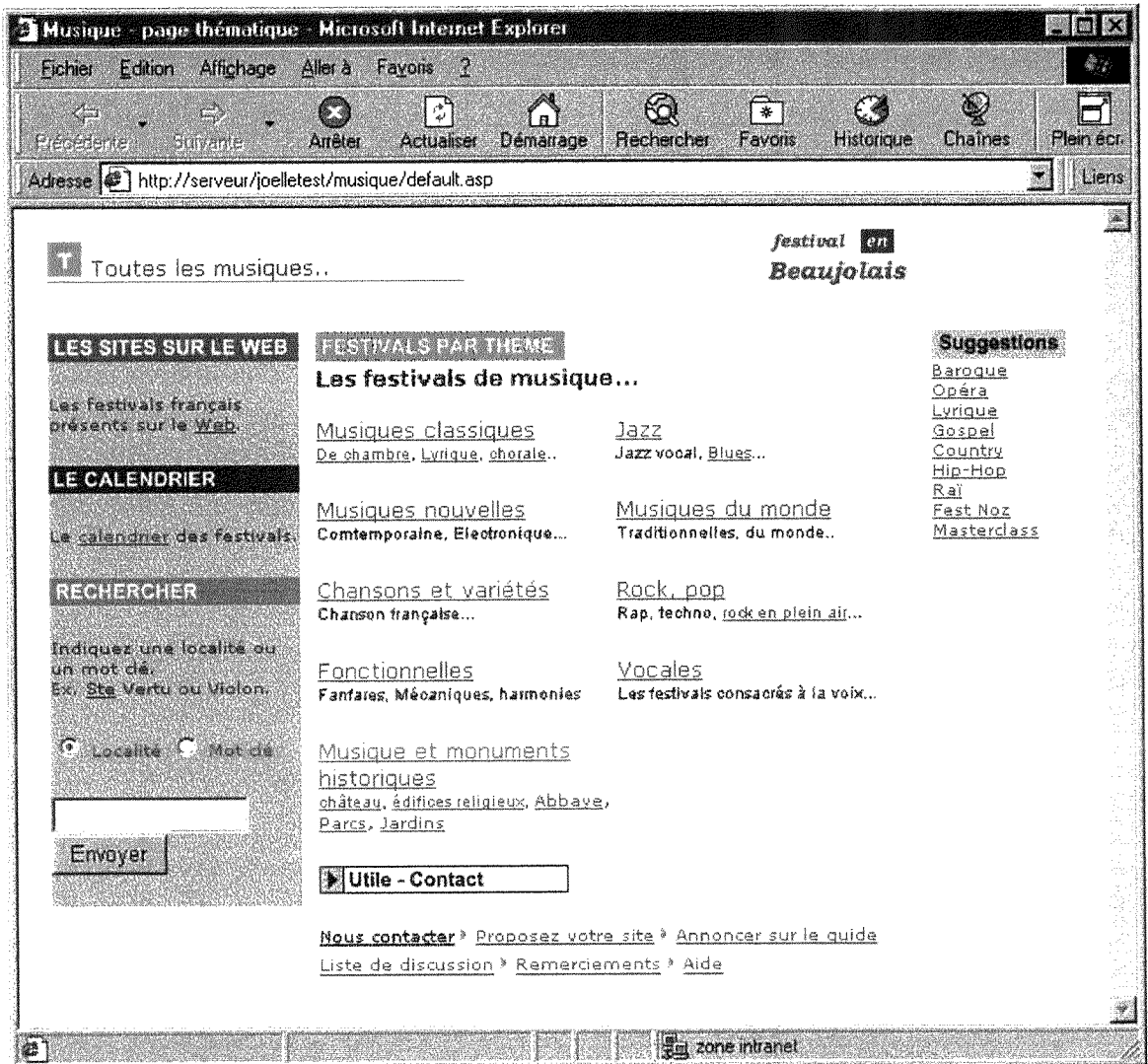


Figure 4. Rubrique festivals de musique existant.

3.1.1. Recherche par ville

D'après les besoins définis, il fallait offrir une recherche par nom de ville. Nous avons donc proposé de faire une liste déroulante avec les noms de villes indexées dans la base de données. Nous avons aussi enlevé le bouton « Envoyer » du formulaire, afin de prendre moins de place dans la page (cf. Figure 6). Pour ceci nous devons faire une requête sur la base de données pour trouver les villes indexées sur le site (environ 1100 villes pour les festivals de musique).

Après une analyse, nous nous sommes rendu compte qu'il fallait éviter de faire la requête plusieurs fois car cela prenait trop de temps. Nous avons alors proposé d'utiliser des **variables session** disponibles avec les pages ASP, lesquelles sont stockées dans le serveur et peuvent être réutilisées pendant la connexion (session) d'un utilisateur avec le serveur.

pour ceci, il a fallu créer un fichier appelé global.asa contenant les informations suivantes :

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
' Initialisation des variables Application
    Sub Application_OnStart
        Application("visitas")=0
    End Sub
' Initialisation des variables session
    Sub Session_OnStart
        Application("visitas")=Application("visitas")+1
        Session("Validation")=0
        Session("Villes")=""
        Session("Nbrevilles")=0
    End Sub
</SCRIPT>
```

Ce fichier doit être placé dans le répertoire racine du serveur pour que les fichiers ASP puissent lire ces informations. Les variables session : Session(«Validation »), Session(« Villes ») et Session(« Nbrevilles ») sont initialisées au début de la session avec l'événement Session_OnStart.

Objectif:

Il s'agit d'afficher toutes les villes indexées dans la BD dans une liste déroulante et d'utiliser les variables session afin ne pas faire deux fois la même requête et ainsi améliorer la performance du site.

Voir les **Actions** la **traduction en pseudo code** et l'**implémentation** de ces actions dans l'ANNEXE A.

3.1.2. Calendrier

Pour mettre au point le calendrier nous avons travaillé avec les pages ASP et le langage VBScript .

Objectif

Faire un calendrier et afficher le mois en cours. Un lien actif permet d'accéder aux jours suivant le jour en cours. De plus, en fin de mois, un lien permet l'affichage du mois suivant, de même, ce lien permet , le premier jour d'un mois, de faire apparaître l'affichage du mois précédent (*cf. Figure 5*).

LE CALENDRIER						
Septembre 1999						
D	L	M	M	J	V	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>	<u>18</u>
<u>19</u>	<u>20</u>	<u>21</u>	<u>22</u>	<u>23</u>	<u>24</u>	<u>25</u>
<u>26</u>	<u>27</u>	<u>28</u>	<u>29</u>	<u>30</u>		
					-->	

Figure 5. Calendrier Développé dans la page actualité de festivals de musique

Ici, une variable Session a été utilisée afin de déterminer si l'utilisateur visite pour la première fois la page de la session en cours. Dans ce cas, nous devons prendre la date du système (Mois, année et jour). Pour naviguer dans les mois suivants, soit le mois et l'année s'incrémenteront (mois suivant et année suivante) soit ils se décrémenteront (mois précédent, année précédente) en fonction du choix effectué, tandis que le jour prendra automatiquement la valeur de 1 (premier jour du mois). Voir les **Actions** et la **traduction en pseudo code** et l'**implémentation** de ces actions dans l'ANNEXE B.

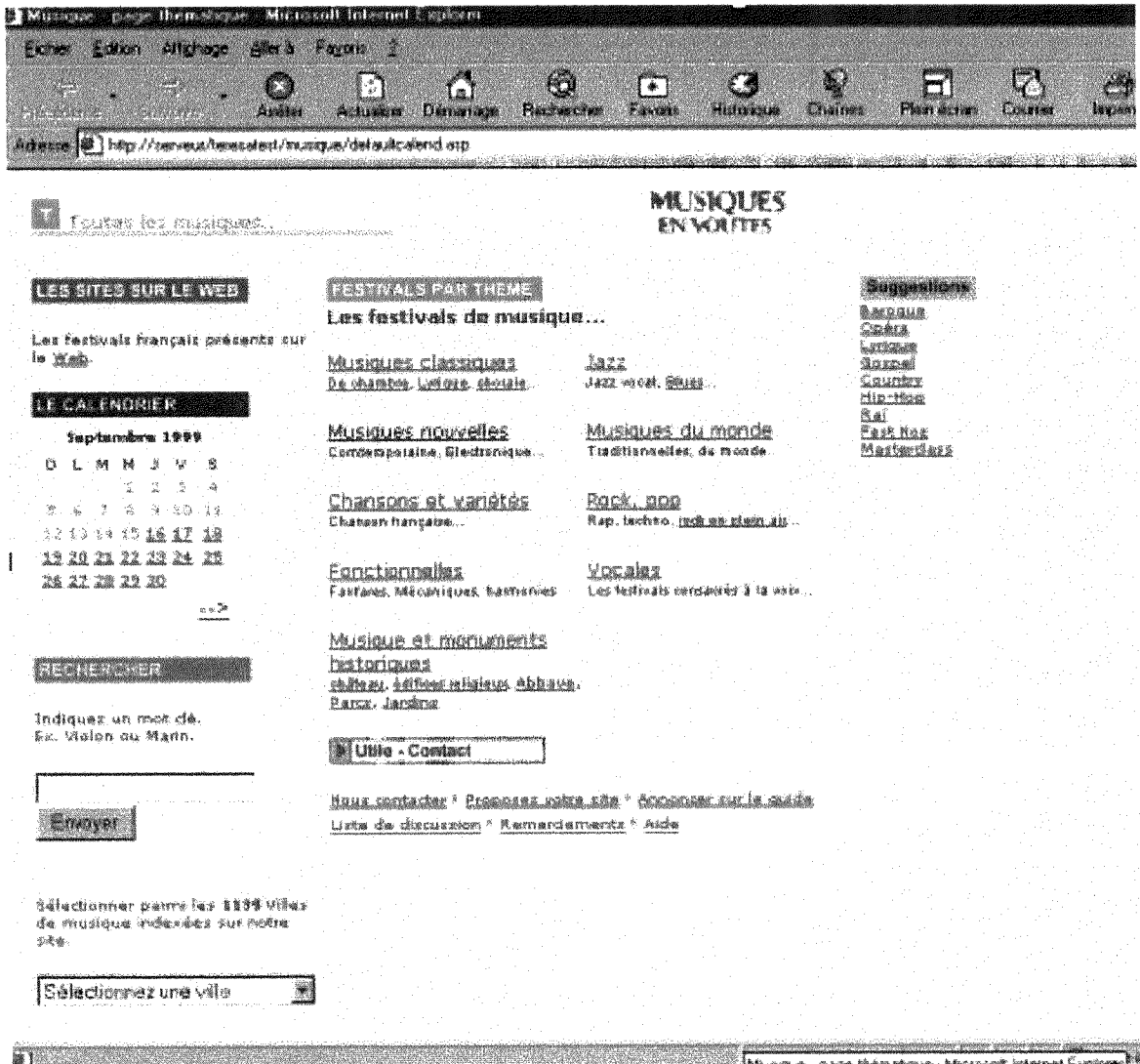


Figure 6. Rubrique festivals de musique modifiée (liste de villes de musique et calendrier).

3.1.3. Une seule page réponse

Lorsque nous avons analysé les pages existantes, nous nous sommes rendu compte que l'on utilisait une page réponse pour chaque requête, nous avons alors proposé d'utiliser une seule page réponse, quelle que soit la recherche choisie : par thème, par ville ou par jour (calendrier).

Pour cela, il a suffi d'envoyer une variable pour chaque rubrique et dépendant de la valeur de celle-ci, la page réponse saura quelle requête SQL devra être exécutée et ainsi afficher les résultats.

Objectif

Afficher les réponses en accord avec une requête posée, dans une seule page réponse.

Actions à développer en VBScript

1. Recevoir les paramètres envoyés par la page de requêtes, parmi lesquels se trouve la variable requête (type de requête)
2. Connexion avec la BD
3. Choisir une requête parmi

 Cas « ville » :

 Requête SQL, réponse 1

 Cas « calendrier »

 Requête SQL, réponse 2

 Cas « theme »

 Requête SQL, réponse 3

 Fin_choisir

4. Afficher les résultats trouvés.

Voir l'**Implémentation ANNEXE C**

3.2. Répertoire thématique des sites de villes.

Dans le but d'enrichir le site , il a fallu effectuer une recherche sur Internet afin de déterminer si le sujet était intéressant et exploitable. Finalement le thème : « les villes et villages touristiques de France qui possèdent un site sur le Web » a été retenu.

Dans un premier temps, il était nécessaire d'identifier les communes de France ayant un aspect touristique. Pour ceci, nous nous sommes servis des guides papier. Ensuite, une fois les villes identifiées, nous avons commencé la recherche sur Internet des adresses URL de ces villes pour ainsi constituer le répertoire des villes. Par ailleurs, il était nécessaire de constituer un thesaurus touristique et de loisirs pour indexer dans la BD les sites des villes. Enfin, nous avons développé l'interface Web des villes touristiques de France.

3.2.1. Recherche des villes touristiques de France

Guides papier

Etant donné le nombre considérable de villes touristiques existant en France, il a fallu limiter la recherche et procéder à une sélection. Pour cela nous avons utilisé des guides papier et les villes déjà indexées dans la base de données existante. Nous avons travaillé avec trois guides papier : un guide des villes et villages « de charme » et deux autres guides qui permettaient d'identifier les villes et villages touristiques de France présentant un intérêt particulier, celui-ci étant exprimé sous forme d'« étoiles ».

3.2.2. Recherche des sites Internet des villes

Une fois les villes identifiées, nous avons commencé la recherche sur Internet. Nous l'avons limitée en fonction du critère suivant : « avoir au moins 10 sites par région ». De cette manière nous pourrions avoir une première version du répertoire de villes, avec une approche globale. Nous avons donc trouvé 328 sites dont 239 villes et villages. Cette résultat s'explique par le fait qu'il était possible d'avoir plus d'un site pour une seule ville ou un village.

Compte tenu de notre sujet de recherche, nous avons décidé de partir de la recherche la plus large vers la recherche la plus précise. C'est-à-dire que nous avons jugé pertinent de commencer par les annuaires de recherche où nous

pourrions trouver une rubrique intéressante qui traitait notre sujet. Dans le cas où notre recherche devenait plus précise, il s'avérait nécessaire d'utiliser des moteurs de recherche.

Utilisation des annuaires sur internet

Annuaire Yahoo :

Comme notre sujet ne traitait que de la France, nous avons décidé de commencer par faire nos recherches avec le moteur français yahoo.fr

Requête	Résultats	Pertinence
Ville +France	8 catégories	Pertinent

Grâce à cette requête nous avons trouvé la rubrique de l'annuaire:

Exploration_geographique→Pays→France→Villes,

(URL : http://www.yahoo.fr/Exploration_geographique/Pays/France/Villes/)

Nous avons alors trouvé une liste de villes classées par ordre alphabétique, proposant, dans la plus part des cas, un ou plusieurs sites Web. Quelquefois il s'agissait de sites officiels mais d'autres fois, de sites personnels. Au cours de cette première phase de recherche, nous nous sommes rendus compte que dans la plupart des cas, l'URL des sites officiels trouvés avaient la structure suivante :

[http://www.ville-\[nom de la ville\].fr](http://www.ville-[nom de la ville].fr)

ou bien :

[http://www.mairie-\[nomde la ville\].fr](http://www.mairie-[nomde la ville].fr)

Cela nous a aidé lors de la recherche plus fine (par ville précise), avec les moteurs de recherche sur Internet.

Au cours de la phase décrite ci-dessus, nous avons aussi trouvé des liens renvoyant à des annuaires ou des sites fédérateurs de sites de villes et de villages de France (*cf ANNEXE D*). Ils proposent, dans plusieurs cas, plusieurs types de recherche (par région, par département, par nom de ville ou de village et

quelquefois par carte). D'ailleurs, avec yahoo.fr nous avons trouvé, dans la catégorie : Références et annuaires, deux nouveaux sites (annuaires) de villes de France. Ceux-ci regroupent des sites Internet de villes de France.

Annuaire	Catégorie	Liens référencés
Yahoo.fr	Références et annuaires : http://www.yahoo.fr/Referencs_et_annuaires/ Annuaires/	http://perso.wanadoo.fr/entreprise/ http://www.annu-ville.com/

Les annuaires trouvés nous ont aidé à poursuivre notre recherche d'une façon plus rapide et efficace. De plus, ces adresses ont été repérées pour continuer le développement et la mise à jour du site.

Utilisation des moteurs de recherche

Altavista

Nous avons utilisé ce moteur de recherche, car parfois le site d'une ville (ou d'un village) n'était pas trouvé avec yahoo.fr. S'agissant d'une recherche précise, il s'avérait nécessaire d'utiliser un moteur de recherche comme Altavista ; pour profiter de la recherche proposée sur l'adresse URL (« url : »). En effet, pour faire les requêtes nous avons utilisé l'expérience acquise, en l'occurrence, nous essayions de trouver le nom de la ville dans l'adresse URL, afin de déterminer si un site officiel existait.

ex.

Requête	Nombre de Résultats	Pertinence
url:annonay AND url : fr Depuis :01/jan/99 Langue : français	6	Pertinent
Pérouge AND url :perouges Depuis : 01/jan/99 Langue : français	26	Pertinent

url:obernai AND url: fr Depuis : 01/jan/99 Langue : français	33	Pertinent
--	----	-----------

3.2.3. Indexation des villes

Au moment de la recherche sur Internet, il nous a fallu travailler avec le logiciel PowerMarks, logiciel d'indexation de favoris et de veille Internet. Il nous a permis d'indexer les sites trouvés en fonction de leur intérêt touristique ; les loisirs proposés, le sport, le patrimoine, la culture, etc..., à la qualité du site (nombre d'informations proposées), à la mise à jour, aux langues dans lesquelles le site est traduit et lorsqu'il existait, au label que porte la ville : ex. Ville d'Art et d'Histoire (VAH).

3.2.4. Thesaurus touristique et des loisirs

Pour bien indexer les villes dans la base de données ACCESS utilisée dans le site, il fallait construire un thesaurus du tourisme et des loisirs, adapté aux besoins de l'entreprise. Pour le faire, nous nous sommes inspirés de la liste d'autorité RAMEAU et du thesaurus DELPHES ainsi que de l'indexation (mots clés) réalisée pour chaque ville avec PowerMarks au moment de la recherche sur Internet.

Une fois le thesaurus construit, il a ensuite fallu travailler avec le personnel de l'entreprise pour l'adapter à ses besoins.

Les grandes rubriques établies dans notre thesaurus de tourisme et de loisirs ont été les suivantes:

Tourisme et Loisirs

(ts)Tourisme culturel

(ts) Patrimoine architectural et jardins

(ts) Gastronomie

- (ts)Artisanat et savoir-faire
- (ts)tourisme historique, scientifique et technique
- (ts)manifestation culturelle
- (ts)musée
- (ts)sites archéologiques
- (ts) loisirs en plein air
 - (ts) Nature
 - (ts)parcs de loisirs
 - (ta) trains historiques
- (ts)Sport et Jeux
 - (ts)Jeux
 - (ts)Sport
 - (ta) manifestation sportive
 - (ts)chasse et pêche
 - (ts) sport de montagne
 - (ts) sport d'hiver
 - (ts) Cyclotourisme et VTT
 - (ts) sport aérien
 - (ts)sport de combat
 - (ts) sport mécanique
 - (ts) randonnées
 - (ts) sport de balle, ballon et boule
 - (ts) sports athlétiques
 - (ts) sport équestre
 - (ts) sport nautique

3.2.5. Réalisation d'une Interface des villes

D'abord, comme dans tout système de Gestion de Base de Données (SGBD), une analyse du MCD s'avère nécessaire, pour ainsi construire notre

Modèle relationnel de Données dans la Base de Données. Ceci afin d'incorporer dans celle-ci nos nouvelles données :

- Les URL de villes
- Le thesaurus de tourisme et de loisirs

Ensuite, la conception de l'interface Web est présentée avec la logique et le code incorporé dans les pages ASP.

3.2.5.1. *Modification de la Base de Données existante.*

Avant d'insérer les sites de villes et de villages, leur information et le thesaurus de loisirs dans la BD, il a fallu bien étudier sa structure ou Modèle Relationnel afin de bien connaître la position et les relations des nouvelles tables par rapport aux tables existantes *voir Chapitre II.2.*

Insertion des nouvelles tables dans de la BD ACCESS.

Pour l'insertion, dans la BD, des nouvelles informations : les sites trouvés sur Internet des communes de France et le thesaurus de tourisme et de loisirs, nous nous sommes inspirés de la méthode Merise. Nous avons donc suivi les démarches suivants :

- ◆ Lister les résultats souhaités
- ◆ Relever les données élémentaires nécessaires aux traitements (à l'exclusion des données calculées)
- ◆ Noter les contraintes sur les données
- ◆ Repérer les entités
- ◆ Attribuer à chaque entité son identifiant et ses propriétés
- ◆ Vérifier qu'à toute valeur prise par l'identifiant ne correspond qu'une valeur de chaque propriété

- ◆ Vérifier que toutes les propriétés d'une entité dépendent directement de l'identifiant.
- ◆ Relations et Cardinalité

Développement :

- ◆ Lister les résultats souhaités

Objet : Faire une rubrique sur Internet des communes touristiques de France.

Application : Loisirs de France sur Internet.

Résultats attendus :

1. Afficher l'adresse URL et les attraits touristiques des communes de France.
2. Indexer les villes et les villages à partir d'un thesaurus de loisirs.

- ◆ Relever les données élémentaires nécessaires aux traitements (à l'exclusion des données calculées)

Données :

Nom de la ville

Adresse URL

Code postal

Langue (Traduction du site)

Caractéristiques du site

Label de la ville

Mots du thesaurus de tourisme et de loisirs

Termes associés

Termes spécifiques

Termes génériques

◆ Noter les contraintes sur les données

Contraintes sur les données :

1. La langue peut prendre une ou plusieurs valeurs.
2. Un site peut être indexé avec un ou plusieurs mots du thesaurus.
3. Un mot du thesaurus peut servir à indexer une ou plusieurs communes.
4. Un mot peut être un terme associé d'un ou plusieurs mots.
5. Une ville ou un village peuvent avoir un ou plusieurs sites (URL)

◆ Repérer les entités

Les entités sont les objets de gestion essentiels du système d'information.

Après une première lecture, deux entités ressortent : les URL VILLES et le THESAURUS Tourisme et Loisirs.

L'entité URL VILLES est un ensemble dont chaque élément est une URL particulière. L'entité THESAURUS Tourisme et Loisirs est un ensemble dont chaque élément est un mot particulier.

Chacune de ces entités possède des caractéristiques. Ainsi une URL Ville est caractérisée par un nom, une adresse URL, Langue, Label, description.

Et le thesaurus loisirs est caractérisé par le mot, les termes associés, les termes spécifiques et les termes génériques.

◆ Attribuer à chaque entité son identifiant et ses propriétés

Il faut maintenant assigner un identifiant à chaque entité pour désigner sans ambiguïté chaque occurrence de l'entité.

A terme, la table Thesaurus Tourisme et Loisirs sera insérée dans la table Thesaurus. En effet, l'entreprise projette de réviser entièrement son outil d'indexation.

Dans le cadre de ma mission, nous n'aborderons pas la partie technique : nous avons créé le Thesaurus Tourisme et Loisirs dans le but de définir les différents thèmes du répertoire des villes.

Pour réaliser les tests, nous avons inséré quelques mots clés du nouveau Thesaurus Tourisme et Loisirs dans la Table Thesaurus (existante).

L'entité URL VILLE apparaît comme suit :

Entité :URL VILLE

Identifiant : numéro URL

Propriétés : nom ville

Adresse URL

Description

Langue

Courrier Electronique

Le Label de la ville a été incorporé à la table Thesaurusgeo, car il n'est pas attribut de la table URL VILLE, mais de la table des Villes (Thesaurusgeo).

Domaine des données :

numéro URL : numérique

nom ville : chaîne de caractères

Adresse URL : chaîne de caractères

Description : chaîne de caractères

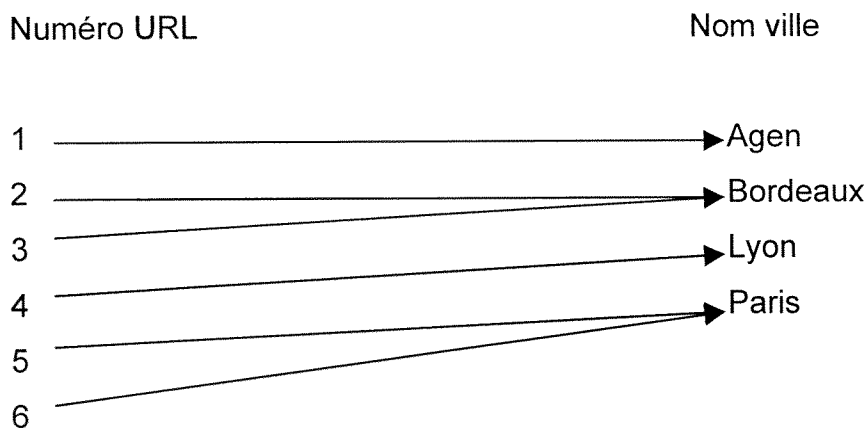
Langue : chaîne de caractères

Courrier électronique : chaîne de caractères

- ◆ Vérifier qu'à toute valeur prise par l'identifiant ne correspond qu'une valeur de chaque propriété (la règle d'énumération).

Le schéma ci-dessous montre qu'à tout élément de l'ensemble de départ (le numéro identifiant) ne correspond qu'un élément dans l'ensemble d'arrivée (nom ville).

Ex.



Cette règle est aussi valide pour les propriétés, adresse URL, description et courrier électronique:

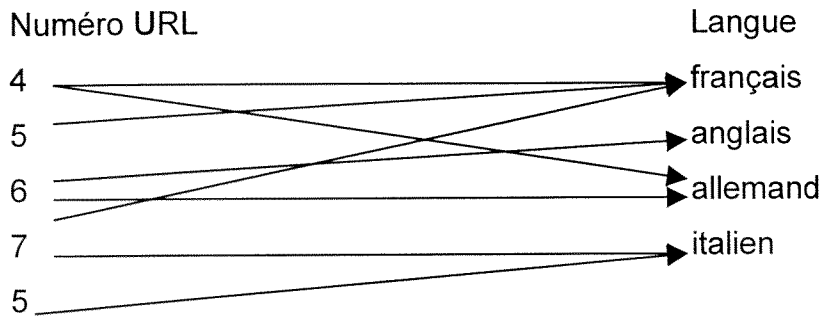
A tout élément de l'ensemble de départ (le numéro identifiant) ne correspond qu'un élément dans l'ensemble d'arrivée (adresse URL).

A tout élément de l'ensemble de départ (le numéro identifiant) ne correspond qu'un élément dans l'ensemble d'arrivée (description).

A tout élément de l'ensemble de départ (le numéro identifiant) ne correspond qu'un élément dans l'ensemble d'arrivée (adresse électronique).

Mais la règle d'énumération ne s'applique pas pour la propriété langue car :

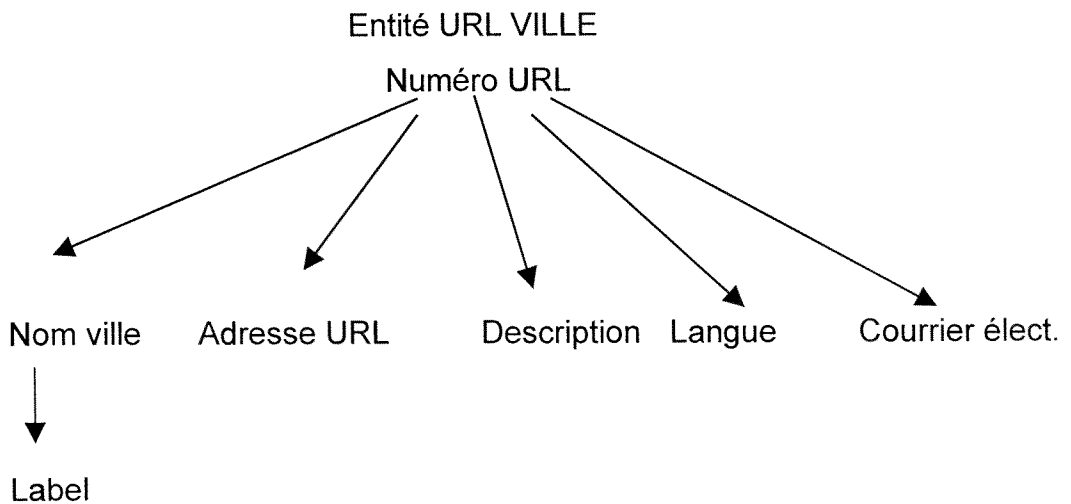
à tout élément de l'ensemble de départ (le numéro identifiant) correspond un ou plusieurs éléments dans l'ensemble d'arrivée (langue).



Nous avons décidé de laisser ce champ multivalué car faire une table uniquement pour la langue ne se justifiait pas véritablement. De plus, cela aurait compliqué la mise à jour .

- ◆ Vérifier que toutes les propriétés d'une entité dépendent directement de l'identifiant.

Pour éviter d'avoir une entité cachée dans une autre, il faut vérifier cette dépendance :



Toutes les propriétés dépendent directement de l'identifiant Numéro URL sauf le Label (ex. ville d'art et d'histoire) qui dépend de la ville. C'est pour cela que nous avons décidé de mettre le Label comme un attribut de la Table Thesaurusgeo (où l'on trouve les régions, les départements et les villes de France).

Domaine

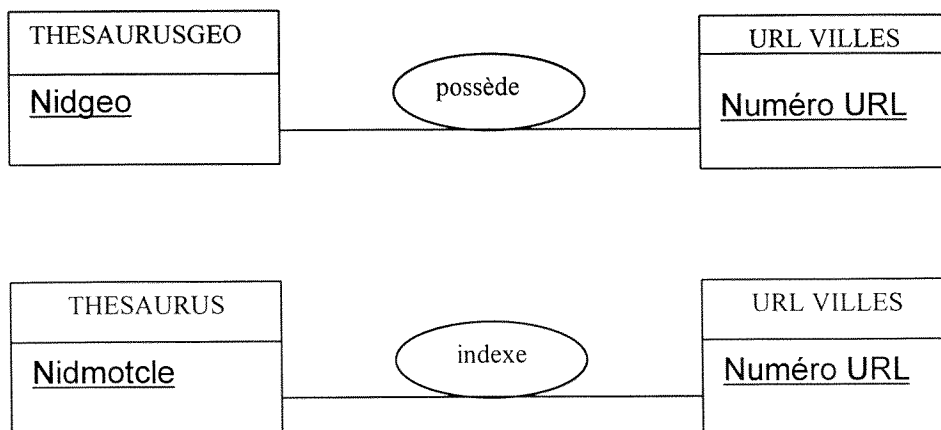
◆ Relations et Cardinalité

Les relations indiquent des liens de sens entre les entités ayant un rapport à la réalité observée.

Par contre la cardinalité d'une entité vis-à-vis d'une relation indique, pour une occurrence d'entité, combien il y a au minimum et au maximum d'occurrences de la relation. De plus les cardinalités indiquent des contraintes entre les entités du modèle, elles sont la représentation de règles auxquelles devra obéir le modèle.

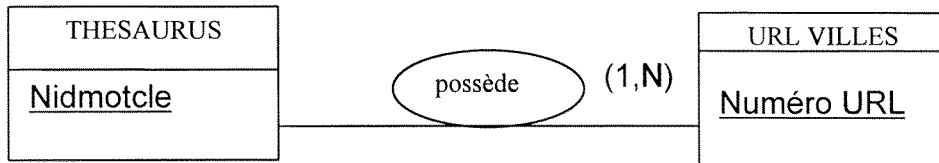
Maintenant, nous pouvons construire les relations et ensuite les nouvelles cardinalités dans un Modèle Entité/Relation.

Tout d'abord, nous savons qu'une commune peut avoir une adresse URL, donc nous construisons notre première relation THESAURUSGEO et URL VILLE.

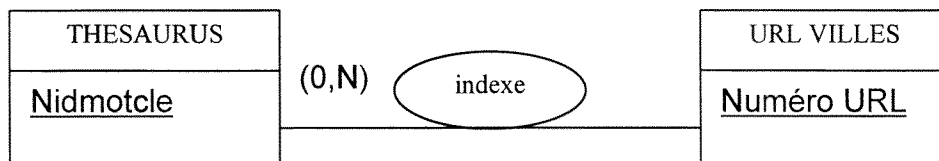


Contraintes entre les entités :

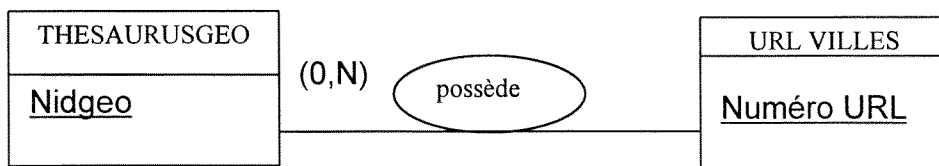
- Un site (URL VILLE) peut être indexé avec au moins un mot et au plus avec N mots du thesaurus(THESAURUS).



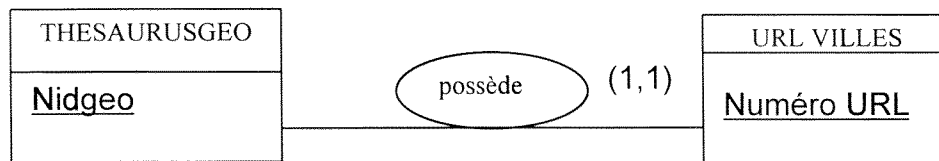
- Chaque mot du thesaurus(THESAURUS) peut servir à indexer plusieurs sites des communes (URL VILLE) mais certains mots peuvent ne pas être utilisés.



- Une ville ou un village (THESAURUSGEO) peuvent avoir plusieurs sites (URLVILLES) mais certaines communes peuvent ne pas avoir de site.



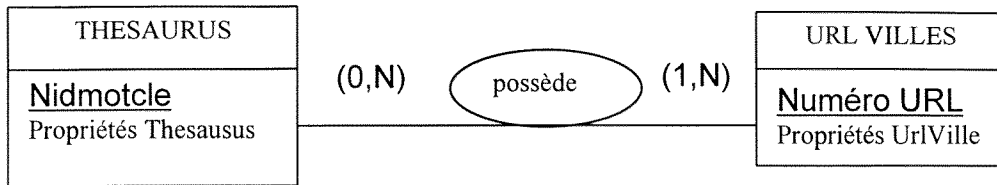
- Un site internet (URL VILLES) appartient, au plus, à une commune (THESAURUSGEO).



Passage au logique

Pour passer du modèle entité/relation au modèle relationnel nous avons fait :

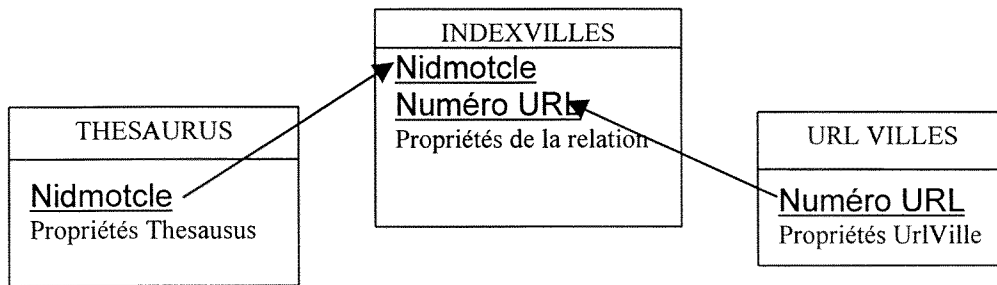
Cas : Relations N/N (1,N/1,N) (1,N/0,N) (0,N/0,N)



Modèle Entité/Relation



Modèle Relationnel

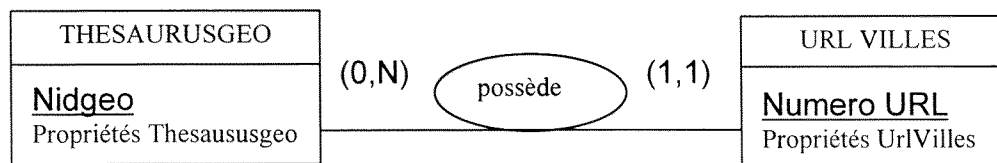


THESAURUS (Nidmotcle, Propriétés Thesaurus)

URLVILLES (Numero URL, Propriétés UrlVille)

INDEXVILLES (Nidmotcle, Numero URL, Propriétés de la relation)

Cas : Dépendance Fonctionnelle



Modèle Entité/Relation



Modèle Relationnel



THESAURUS (**Nidgeo**, Propriétés Thesaurusgeo)

URLVILLES (**Numéro URL**, Propriétés UriVille, Nidgeo)

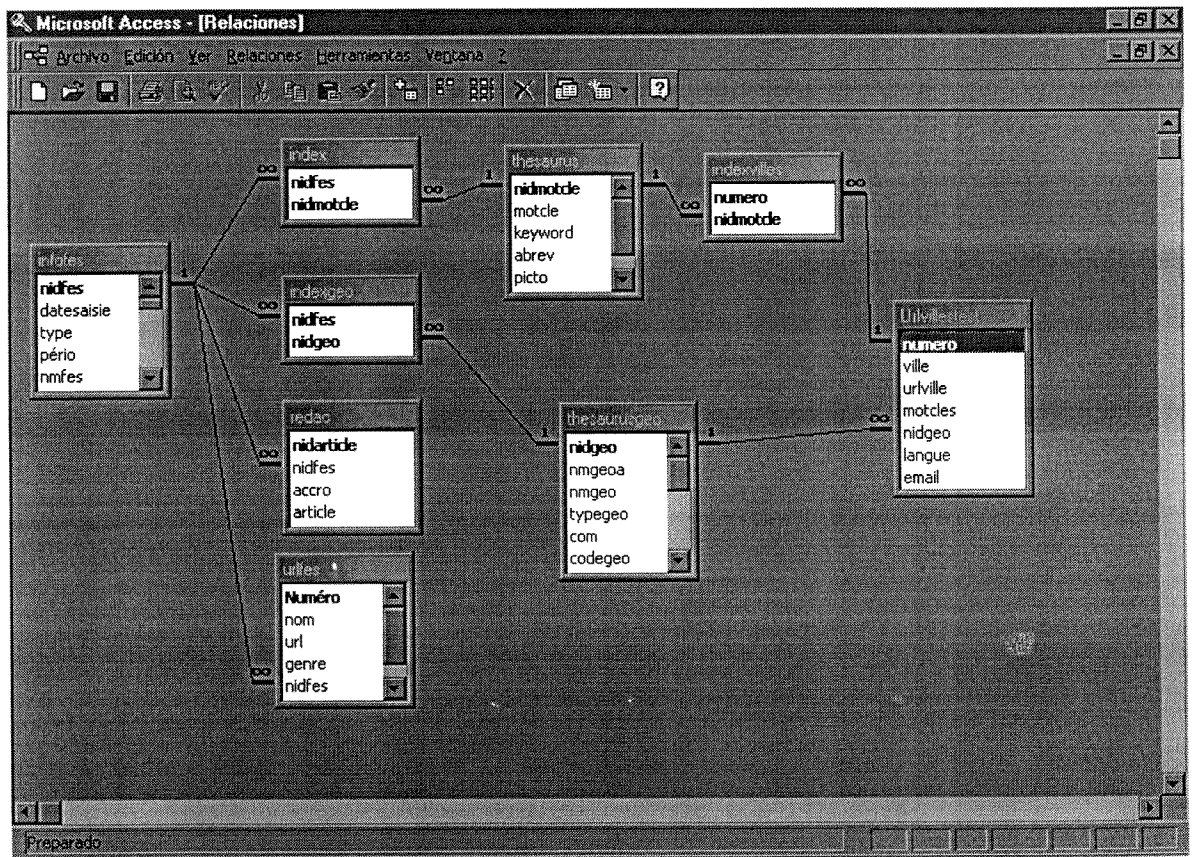


Figure 7. Modèle Relationnel définitif

3.2.5.2. Conception de la rubrique des villes

Pour réaliser la page Web des sites des villes touristiques de France, nous avons proposé de rendre accessible aux utilisateurs deux types de recherche :

- une recherche par thème (indexation des villes à l'aide des mots du thesaurus)
- une autre recherche par nom de la ville, avec option permettant de situer la ville sur une Carte.

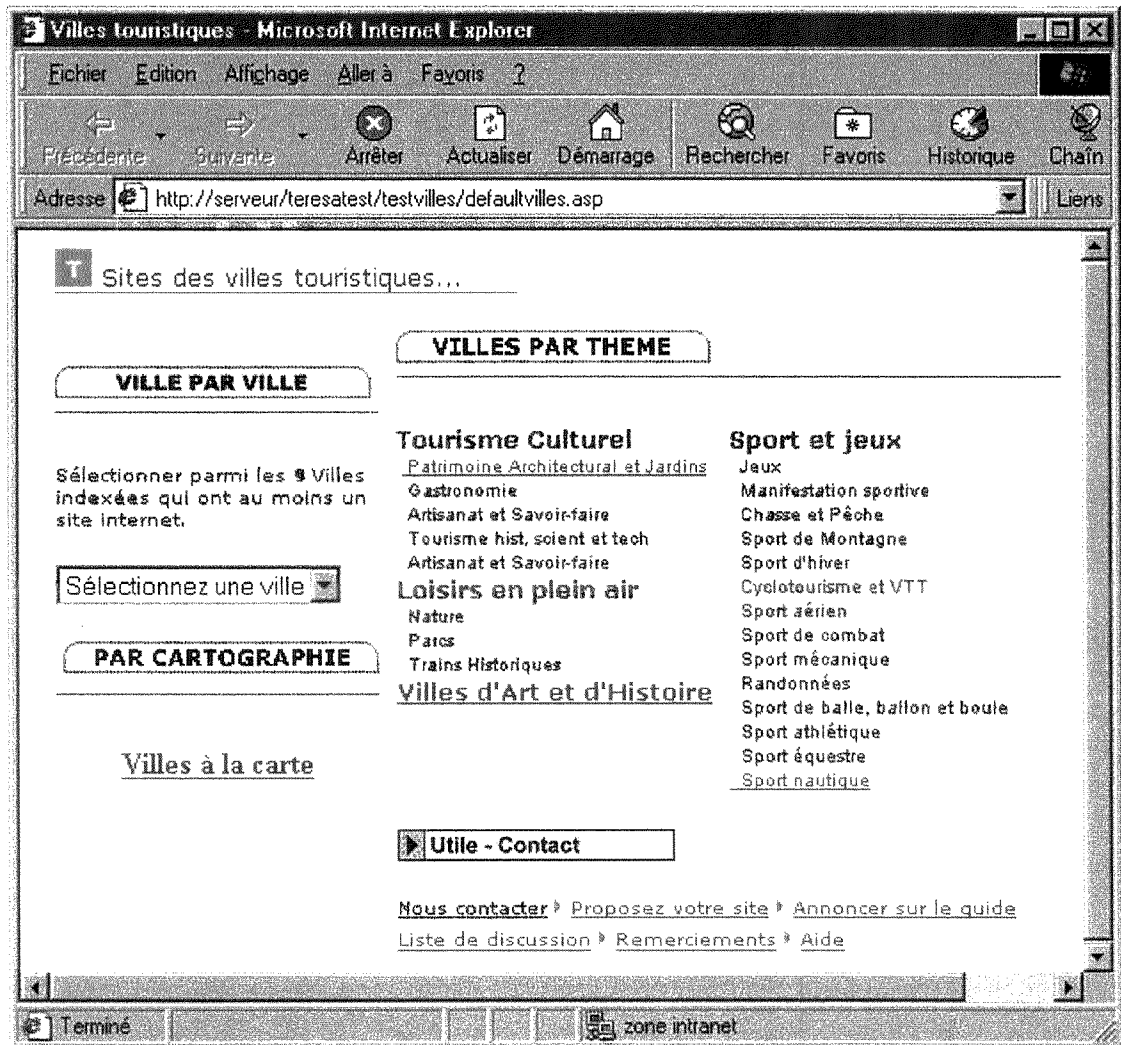


Figure 8. Rubrique villes touristiques (recherche par thème et par ville, avec une approche cartographique).

Remarques :

Pour développer cette partie, il a fallu travailler avec les pages ASP (Active Server Pages), le langage HTML, VBScript, Java et le langage de manipulation des données SQL. D'autre part, nous ne nous sommes pas occupés de la charte

graphique et de l'arborescence logique car c'était déjà un projet envisagé par l'entreprise.

Recherche par nom ville

Dans cette rubrique, nous avons fait une requête pour compter le nombre de villes qui ont au moins un site Internet (239): « Sélectionnez une ville parmi les X villes indexées sur notre site qui ont au moins un site Internet. (cf Figure 8).

Page demande d'informations :

Algorithme utilisé et mise en œuvre.

Objectif

Compter le nombre de villes qui ont au moins un site Internet et afficher les noms de villes dans une liste déroulante.

- Voir les **Actions** et la **traduction en pseudo code** et l'**implémentation** de ces actions dans l'ANNEXE E

Page réponse :

Algorithme utilisé et mise en œuvre

Objectif

Répondre à l'utilisateur selon la ville choisie

Voir les **Actions** et la **traduction en pseudo code** et l'**implémentation** de ces actions dans Voir ANNEXE F.

Recherche thématique

Page de demande d'informations :

Objectif

Afficher les villes par thème.

Actions :

1. Avec l'aide du thesaurus loisirs que nous avons déjà construit, nous avons mis les thèmes principaux dans cette rubrique.

Tourisme culturel

Patrimoine architectural et jardins

Gastronomie

Artisanat et savoir-faire

tourisme historique, scientifique et technique

manifestation culturelle

loisirs en plein air

Nature

parcs de loisirs

trains historiques

Sport et Jeux

Jeux

manifestation sportive

chasse et pêche

sport de montagne

sport d'hiver

Cyclotourisme et VTT

sport aérien

sport de combat

sport mécanique
randonnées
sport de balle, ballon et boule
sport athlétique
sport équestre
sport nautique

Nous avons aussi décidé d'insérer un thème qui nous a semblé très intéressant : le label, c'est-à-dire une appellation attribuée à certaines villes. Dans notre cas nous avons repéré les « villes d'art et d'histoire »

2. Comme nous avons attribué un numéro correspondant à chaque mot du thesaurus, nous l'avons fixé afin de faire la requête à partir de la rubrique choisie.

Ex. lien du thème Patrimoine Architectural

```
<a href="http://general/rqville.asp?mots=210500">Patrimoine Architectural et Jardins</a>
```

Dans l'exemple choisi, le numéro dans le thesaurus est 210500 ; la page appelée rqville.asp prendra ce paramètre pour afficher les résultats d'une requête contenant ce numéro.

Page réponse :

Algorithme et mise en œuvre

Objectif

Afficher les villes dont l'utilisateur a besoin

Voir les **Actions**, la **traduction en pseudo code** et l'**implémentation** de ces actions dans l'**ANNEXE G**.

Dans le cas où l'utilisateur choisit la rubrique « villes d'art et d'histoire », au lieu d'envoyer un numéro, nous envoyons une chaîne avec le label (dans ce cas le label est: « VAH »).

Ex. lien du thème Villes d'art et d'histoire

```
<a href="http://general/rqville.asp?label=VAH"><b>Villes d'Art et d'Histoire</b></a>
```

Objectif

Afficher les villes d' « Art et d'Histoire »

Voir les **Actions**, la **traduction en pseudo code** et l'**implémentation** de ces actions dans l'ANNEXE H.

Cartographie

La recherche est traitée dans le chapitre suivant, où nous présenterons différentes techniques pouvant être utilisées pour l'insertion de la carte géographique dans le site.

3.3. Approche Cartographique

L'insertion d'une carte géographique sur le site des festivals était devenue une nécessité. En effet, la cartographie permet de connaître la situation géographique des festivals se déroulant en France bien entendu, mais aussi de situer les villes touristiques de France. Cette stratégie semble très intéressante et importante au niveau de la valeur ajoutée à l'information proposée mais aussi de la compétitivité du site par rapport aux concurrents. De plus, avec l'approche cartographique, le service rendu aux utilisateurs est notablement amélioré.

Il fallait donc trouver et comprendre une méthodologie économique à exploiter au sein de l'entreprise et adaptée à ses besoins.

D'abord, quelques techniques cartographiques « statiques », par rapport aux coordonnées utilisées, sont analysées. Ensuite, d'autres techniques plus « dynamiques », essayant d'exploiter le nouveau logiciel dans l'entreprise : MapInfo, sont présentées.

3.3.1. Quelques Solutions « Statiques »

3.3.1.1. HTML

Tout d'abord nous avons cherché les techniques utilisant le langage HTML. Nous avons trouvé la balise MAP avec son attribut USEMAP. Dans ce cas le client (navigateur) interprète la carte sans accéder à nouveau au serveur. De plus, l'image utilisée doit être une image au format gif ou jpg et les coordonnées de chaque zone (lien) doivent être prédéterminées. Ce qui rend la carte statique.

ex :

```
<IMG SRC= « mape.gif » USEMAP= « #Carte »>
<MAP NAME= « Carte »>
  <!--Zone rectangle (lien) qui pointe vers le document provence.html -->
  <AREA SHAPE= « RECT » HREF= « provence.html » COORD=
  « 140,20,280,60 »>
  <!--Zone Polygone : triangle (lien) qui pointe vers le document rhone.html -->
  <AREA SHAPE= « POLY » HREF= « provence.html » COORD=
  « 100,100,180,80,200,140 »>
  <!--Zone Cercle (lien) qui pointe vers le document paris.html -->
  <AREA SHAPE= « RECT » HREF= « provence.html » COORD=
  « 80,100,160 »>
</MAP>
```

3.3.1.2. Image réactive

Pour créer une image réactive, il est nécessaire d'utiliser 3 documents et un programme (en « C ») qui reçoit un fichier, lequel contient le chemin d'accès au document lui-même contenant les liens et coordonnées de l'image utilisée.

La balise utilisée est ISMAP :

```
<A HREF= « URL »><IMG SRC= « carte.gif » ISMAP></A>
```

- L'URL est considérée comme un script CGI.
- La requête est caractérisée par les coordonnées du point de l'image sur lequel la personne a cliqué.
- L'utilisateur calcule les coordonnées du pointeur de la souris par rapport au sommet gauche de l'image et ces coordonnées (x,y) sont concaténées à l'URL.

Ex.

X=50, y=18 → URL : <http://machinehost/cgi-bin/imagemap/ligas ?50,18>

Exemple avec la balise ISMAP :

```
<A HREF= « http://machinehost/cgi-bin/imagemap/liens »>
<IMG SRC= « carte.gif » ISMAP></A>
```

Où « liens » est défini dans le fichier de configuration et montre le chemin du fichier liens.map (fichier texte), lequel contient les liens de l'image avec ses coordonnées respectives.

Ex :

```
#lien par default
http://machinehost/Default/default.html
#Region Provence
http : // machinehost/carto/provence.html x1,y1,x2,y2
#Paris
http : // machinehost/carto/paris.html x3,y3,x4,y4
```

3.3.1.3. HTML Dynamique, Active X

Une autre option consistait en l'emploi de HTML Dynamique avec la balise OBJECT pour l'utilisation des objets ActiveX et un script en JavaScript pour gérer l'événement « click » de la souris . Cette fois il fallait utiliser 2 fichiers. Un fichier HTML et un fichier texte comportant dans chaque ligne l'information concernant chaque ville.

Dans le fichier HTML, il est nécessaire d'indiquer le nom du fichier source (txt) qui contient l'information de chaque image gif ou jpg ainsi que la position où sera placée l'image.

Ex : Fichier html

```
<html>
<head>
<title>
Essai cartographique
</title>
</head>
<body>
<p id=region>Cliquez sur un état <br>
Afin de voir <br>
sa population<br>
et sa géographie. </p>
<p>
<object id="statelist" classid="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83" border=0
width="0" height="0">
<param name="DataURL" Value="info.txt">
<param name="UseHeader" Value="True">
</object>
<p>

```

```



<table><tr><td>

</td><td>
<span id=NOM Datasrc=#stalist Datafld="NOM"></span> est une ville. Son code:
<span id=CODE_DEPART Datasrc=#stalist Datafld="CODE_DEPART"></span>. Sa population
est de
<span id=POPSDC Datasrc=#stalist Datafld="POPSDC"></span> habitants. Sa superficie:
<span id=SUPERFICIE_KM2 Datasrc=#stalist Datafld="SUPERFICIE_KM2"></span>
</td></tr>
</table>
<script language=JavaScript>
    var srcElement

    function documentClick()    {
srcElement = window.event.srcElement;
        if (srcElement.id=="Drome")    {
            statelist.recordset.AbsolutePosition = 1;
        }else    {
            if (srcElement.id=="Isere")    {
                statelist.recordset.AbsolutePosition = 2;
            }else    {
                if (srcElement.id=="Paris")    {
                    statelist.recordset.AbsolutePosition = 3;
                }
            }
        }
        Picture.src=srcElement.id + ".gif";
    }
    document.onclick = documentClick;
</script>

```

```
</table>
</body>
</html>
```

Contenu et structure du Fichier info.txt :

```
"CODE_DEPART","NOM","POPSDC","SUPERFICIE_KM2"
"26","DROME",414072,6559.97
"38","ISERE",1016228,7860.57
"56","Paris",89808,988.90
```

Pour l'entreprise, ces dernières solutions semblaient trop statiques au niveau du positionnement des points dans une image (carte).

De plus, l'entreprise a décidé de faire l'acquisition du logiciel (Système d'information géographique) MapInfo qui permet de dessiner des cartes géographiques vectorielles avec pour chaque figure (de type point, polygone, lignes etc...), des informations associées (nom, population, superficie etc...) stockées dans une table. Cette acquisition a permis d'envisager d'autres solutions.

3.3.2. Solutions dynamiques

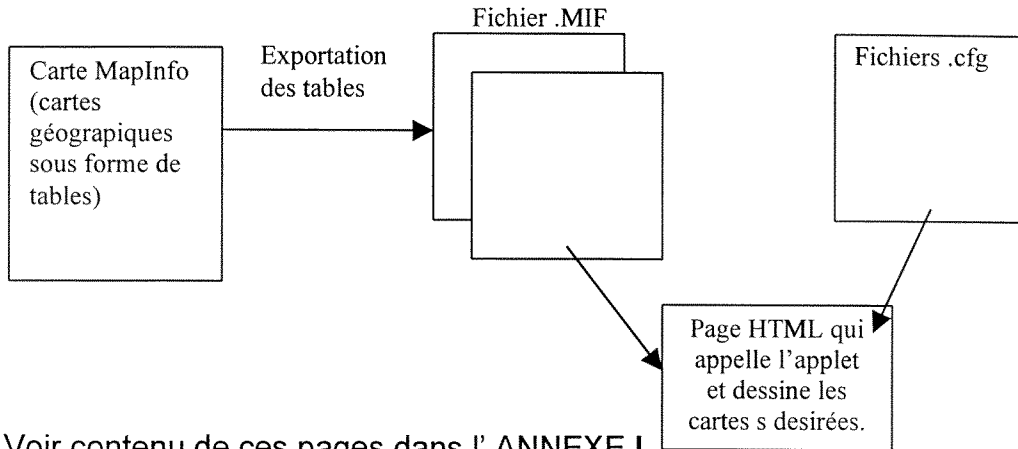
3.3.2.1. MapInfo et NetMaps

Au cours d'une recherche préliminaire, le logiciel NetMaps a été trouvé sur Internet. Celui-ci permet d'afficher sur le Web des cartes au format MapInfo. Une version « demo » de ce logiciel a été téléchargé afin d'analyser son fonctionnement.

Après étude de la documentation du logiciel NetMaps, il s'est avéré nécessaire de:

- exporter les cartes vectorielles MapInfo dans un format d'échange .MIF et .MID ;
- générer un fichier .cfg indiquant le nombre de couches (c'est-à-dire, une couche pour les régions, une autre pour les départements, une autre pour les villes etc...), les noms des fichiers .MIF et .MID et les couleurs des objets, entre autres.

- créer une page html (cf Figure 8, dans la partie « villes à la carte ») qui appelle un applet (NetMaps2.class) et a besoin d'envoyer comme paramètre le nom du fichier généré .cfg.



Voir contenu de ces pages dans l' ANNEXE I.

Rq : pour exporter une carte MapInfo au format .MIF et format .MID, il faut aller dans le menu Table→Exporter→choisir format .MIF → Sauvegarder (cf. Figure 9).

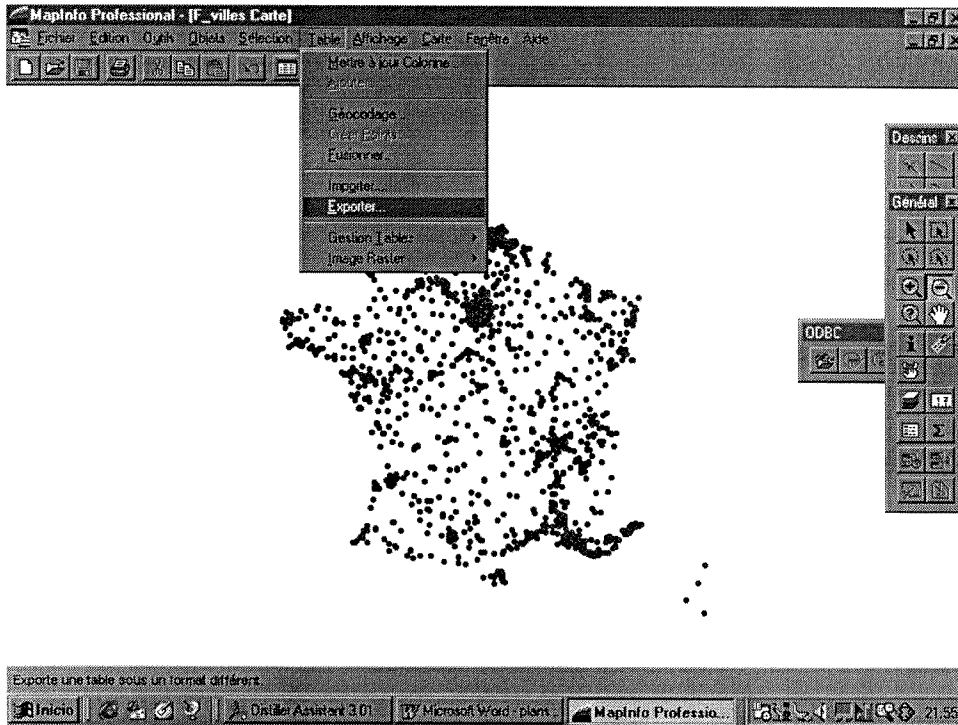


Figure 9. Exportation d'une carte MapInfo au format .MIF et format .MID.

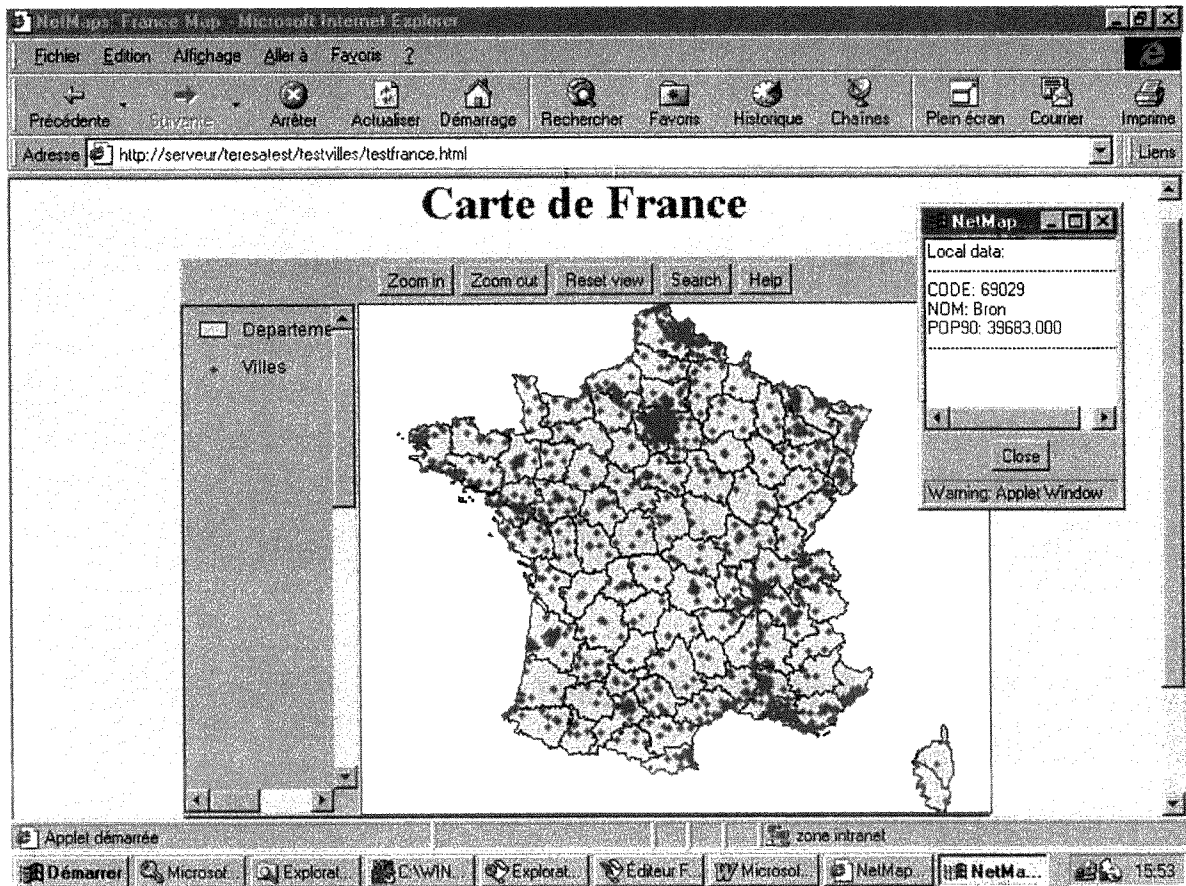


Figure 10. Page générée par la technique NetMaps (applet) ; deux couches : les Départements et les Villes de France.

Si cette solution permet d'obtenir une carte interactive, elle présente cependant certaines limites :

en effet, NetMaps permet d'afficher l'intégralité des fichiers MIF et MID. Dans le cas de fichiers importants, le chargement peut prendre un certain temps. L'applet NetMaps ne permet pas de sélectionner en amont ce que l'on souhaite afficher. On ne peut donc pas générer dynamiquement une carte en fonction de critères précis demandés par l'utilisateur.

3.3.2.2. *Les Servlets*

Une autre solution envisageable était l'utilisation des servlets programmés en Java, car nous savions que d'autres sites (et même d'autres concurrents) travaillaient avec cette technologie.

Tout d'abord, il fallut comprendre ce qu'était un servlet et comment il fonctionnait.

Nous avons donc commencé nos recherches sur le site de java.sun.com et sur le site de gamelan.com. Par ailleurs, la liste de discussion : listserv@java.sun.com s'est avérée être une source d'information très utile.

Les servlets sont la version, côté serveur, d'une applet : un élément de code Java qui peut être chargé par un serveur Web et utilisé pour gérer les requêtes client (comme le fait le CGI). C'est une technique nouvelle et Sun voudrait en faire le nouveau paradigme de la programmation pour les serveurs Web. Les servlets sont persistants, indépendants de la plate-forme et incorporent toutes sortes de caractéristiques avancées dont la sécurité, un accès facile aux bases de données et une intégration beaucoup plus facile avec les applets Java.

Les servlets sont un moyen d'étendre les fonctionnalités des serveurs. Par exemple, les servlets peuvent étendre les possibilités d'un serveur Web de la même manière que le font les scripts CGI. Mais, les servlets sont moins coûteux en ressources que les scripts CGI. Comme les servlets sont entièrement écrits en Java ils sont, à la différence des scripts CGI, multi-plateformes.

Pour pouvoir utiliser cette technologie il a été nécessaire de télécharger différents logiciels :

- jdk2.1 qui contient des programmes(comme par exemple javac.exe ou java.exe) qui servent à compiler et à exécuter, respectivement, des programmes écrits en java.

-jsdk2.1 qui est un serveur de servlets (moteur) pour développer et tester des servlets.

-jsdk2.0 car la version 2.1 ne contenait pas les classes jsdk.jar. Ces classes sont importantes car elles contiennent les classes des servlets.

Configuration du serveur

Avant de commencer les tests avec les servlets, il a été nécessaire de configurer la machine en additionnant d'autres valeurs aux variables d'environnement : PATH et CLASSPATH, dans le fichier Autoexec.bat pour un PC ou, pour un serveur NT dans : Panneau de configuration → système → Variables d'environnement.

Ex. Autoexec.bat

Pour le répertoire des programmes exécutables (compilateur, déboguer etc.) :

```
SET PATH=[...]JDK1.2.2\BIN
```

Pour le répertoire des classes

```
SET CLASSPATH=.;[...]Jsdk2.0\lib\jsdk.jar
```

Ensuite, pouvoir exécuter les servlets il faut démarrer le serveur de servlets avec la commande :

```
[...]jsdk2.1\ > startserver
```

Le répertoire de servlets est : <http://localhost:8080/examples/servlet/> (où servlet est le surnom de Web-inf\servlets). Il est nécessaire de placer les servlets dans ce répertoire pour les exécuter.

Enfin, pour éteindre le serveur de servlets il faut exécuter la commande :

```
[...]jsdk2.1\ > stopserver
```

Principe de fonctionnement du servlet

Les paramètres envoyés par le formulaire HTML (cf. Figure 11) sont récupérés et

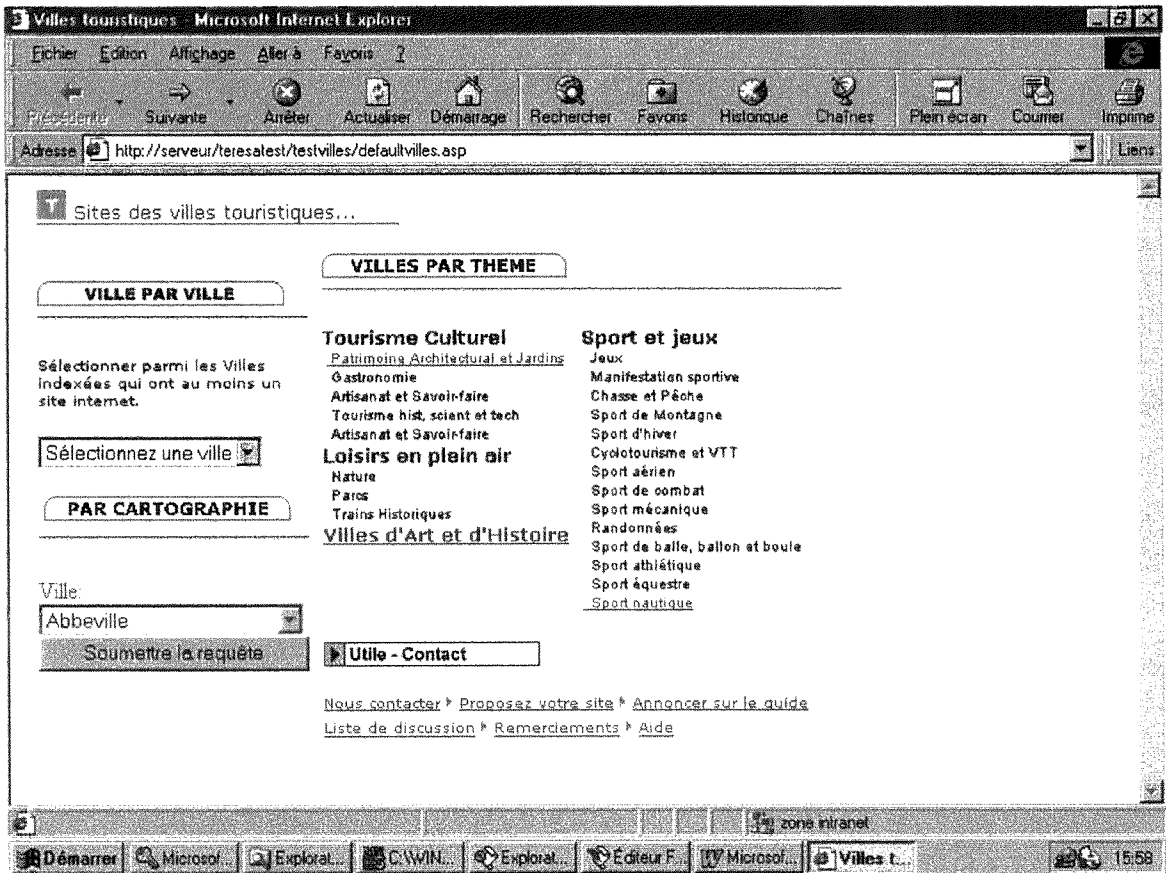
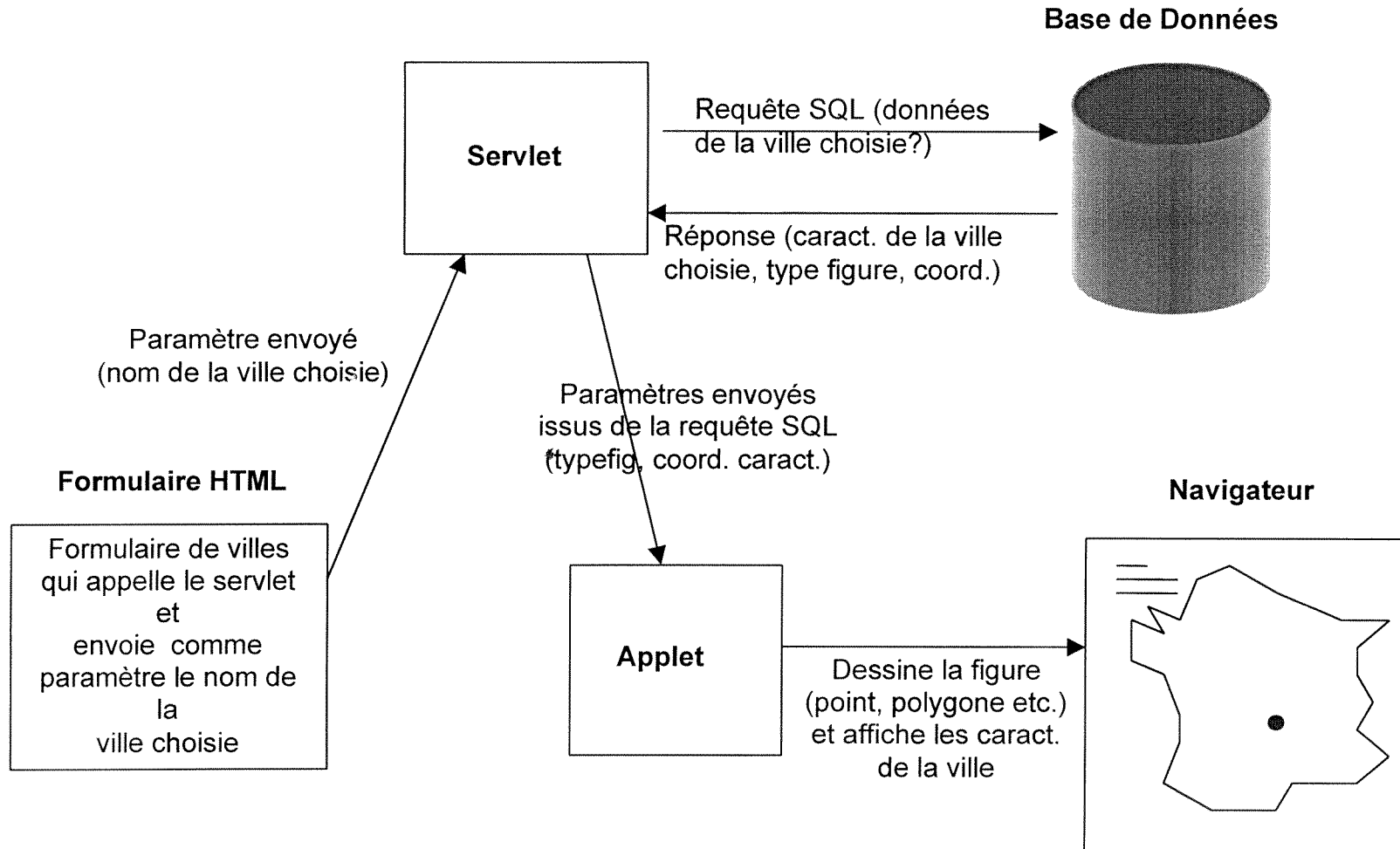


Figure 11. Formulaire Html qui appelle le Servlet (dans la partie Par Catographie)

insérés , au niveau du servlet dans la requête SQL. Le servlet établit ensuite la connexion à la base de données et exécute la requête. Enfin, les résultats sont transmis à l' applet qui est chargée d'afficher les résultats (cf. Figure 12). Ce principe de fonctionnement est représenté dans le schéma suivant :

Fonctionnement



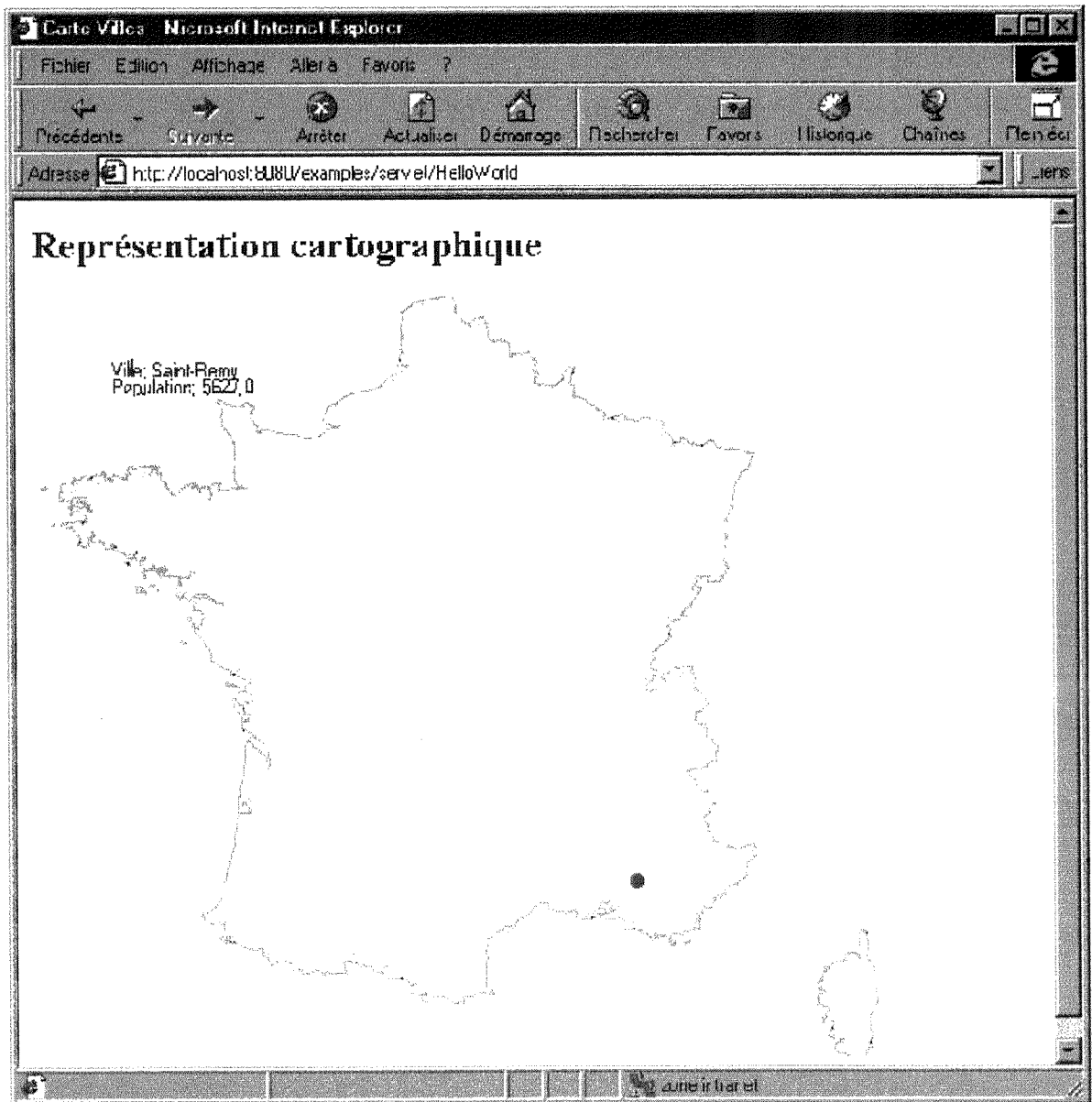


Figure 12. Page qui est construit par le servlet et qui interroge une base de données pour envoyer les résultats vers un applet qui affiche les résultats.

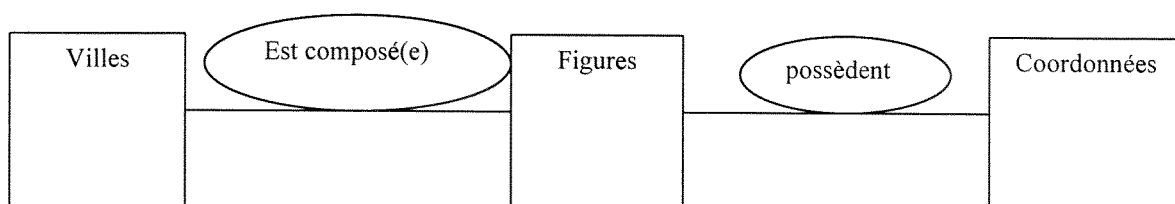
Base de données

La base de données à été générée à partir des fichiers Mapinfo exportés au format MIF et MID (cf page 42 et 43 et ANNEXE I).

Les fichiers MIF et MID n'étant pas exploitables tels quels, il a fallu d'abord, faire un programme en C pour les convertir en fichiers texte (champs et valeurs séparées par des virgules) pour pouvoir ensuite importer ces derniers dans des tables Access.

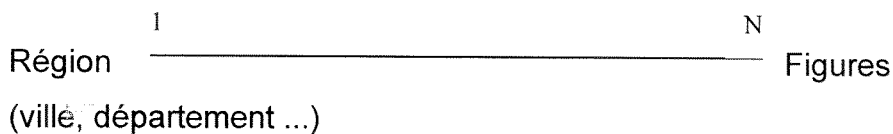
Nous avons proposé pour la base de données la structure suivante :

Relations :



Cardinalité :

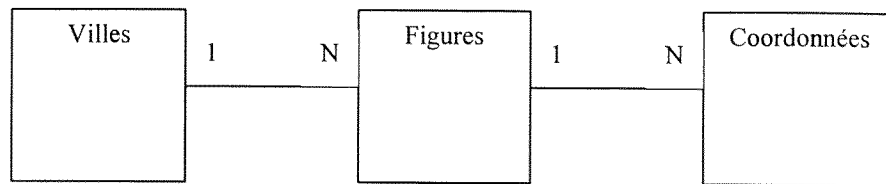
Chaque ville, peut être représenté par une ou plusieurs figures (de type point, polygone etc.).



Chaque figure est formée d'un (figure de type point) ou de plusieurs points (figure de type polygone, ligne etc.).



Enfin, le modèle relationnel a été défini comme suit (cf. Figure 13):



Où, les attributs de chaque entité sont les suivants :

Entité : Villes

Identifiant : code INSEE

Propriétés : nom de la ville

Population

Entité : Figures

Identifiant : Clé figure

Propriétés : codeVille(INSEE) (clé secondaire)

typefigure

nombre de points

Entité : Coordonnées

Identifiant : clé coordonnées

Propriétés : clé figure (clé secondaire)

Coordonnée X

Coordonnée Y

Note : pour faire nos tests nous avons travaillé avec cette base à part. Pour la continuation de ce projet il faudra relier ces tables à la base de données du site.

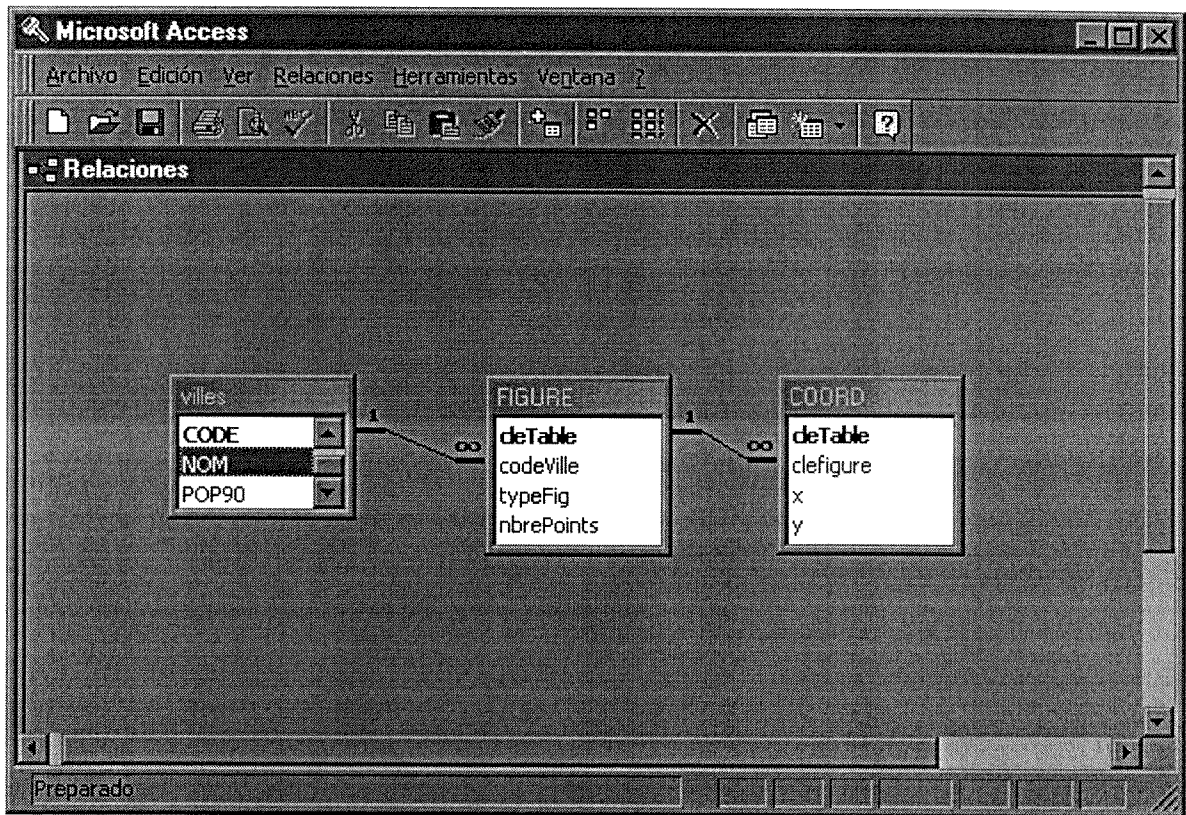


Figure 13. Base de Données créée à partir des fichiers MIF et MID (villes de France).

Pour le code du servlet, de l'applet et du formulaire qui appelle le servlet, voir l'ANNEXE J.

Bilans :

En conclusion, parmi les solutions présentées, les plus adaptées aux besoins et aux contraintes de l'entreprise pour exploiter sur internet des cartes géographiques semblent être : le logiciel Netmaps et les servlets.

NetMaps est une solution économique (en terme de temps et de ressources) et facile à intégrer au sein de l'entreprise. Les servlets sont un outil très puissant qui pourraient inspirer pour le développement d'une application encore plus complète, personnalisée et tout à fait adaptée aux nécessités requises.

Conclusion

Je peux donc dire que le projet réalisé pendant le stage a été très enrichissant pour moi. Tout d'abord, il m'a permis de développer des pages Web dynamiques, avec une technique autre que HTML, c'est-à-dire des pages ASP. De plus, j'ai pu mettre en pratique des connaissances acquises à l'Enssib, comme la création d'un thesaurus et notamment les démarches pour tout SGBD.

Par ailleurs, il a contribué à acquérir un esprit de coopération, de communication et de solidarité dans un groupe de travail. Mais aussi, dans certains cas, il m'a permis d'être plus autonome.

D'autre part, ce stage m'a apporté de nouvelles connaissances et techniques que je méconnaissais auparavant. Les servlets, étant un nouveau paradigme de la programmation, offrent d'autres perspectives pour la programmation sur Internet tout en possédant les avantages du langage java ; la sécurité, l'indépendance de la plate-forme, la facilité d'incorporation et la robustesse.

Enfin, je pense que le projet réalisé m'a apporté de nouvelles ouvertures dans ma vie professionnelle.

ANNEXES

ANNEXES

ANNEXE A : MODIFICATION RÉALISÉE : RECHERCHE PAR VILLE.....	55
ANNEXE B : MODIFICATION RÉALISÉE : RECHERCHE PAR JOUR.....	62
ANNEXE C : UNE PAGE RÉPONSE (THÈME, VILLE OU CALENDRIER).....	74
ANNEXE D : ANNUAIRES DE VILLES DE FRANCE.....	76
ANNEXE E : RUBRIQUE DES VILLES : RECHERCHE PAR VILLE.....	77
ANNEXE F : UNE PAGE RÉPONSE, RECHERCHE PAR VILLE.....	82
ANNEXE G : UNE PAGE RÉPONSE, RECHERCHE PAR THÈME.....	85
ANNEXE H : UNE PAGE RÉPONSE, VILLES D'«ART ET D'HISTOIRE».....	88
ANNEXE I : FICHIERS UTILISÉS AVEC LE LOGICIEL NETMAPS.....	91
ANNEXE J : CODE UTILISÉ POUR LA CONCEPTION DES SERVLETS.....	95

ANNEXE A : Modification réalisée : Recherche par ville.

Actions

1. Faire la connexion avec la Base de Données.
2. Si c'est le début de la session, c'est-à-dire : Session(« Validation »)=0
 - 2.1 Faire une Requête SQL, joindre 4 tables de la BD (index, infofes, thesaurusgeo et indexgeo) pour trouver toutes les villes(nmgeo) et leur clef (nidgeo) correspondant à un certain type de festivals (p.ex. les festivals de musique où la clé (nidmotcle) est égale à 10000).
Ordonner le résultat par nom de ville.
 - 2.2 Exécuter la requête et garder le résultat dans une variable (RS1)
 - 2.3 Initialiser le compteur de villes à zéro (k=0)
 - 2.4 Tant qu'il existe encore des éléments dans la variable RS1
 - Incrémenter en 1 la variable k
 - Avancer vers l'élément suivant (RS1.next)
 - Fin_tant
 - 2.5 Fermer et vider la variable RS1
 - 2.6 Affecter à la variable Session(« nbrevilles »), la valeur k
3. Afficher le nombre de villes
 - Ecrire, le nombre des villes est : Session(« nbrevilles »)
4. Si la variable Session(« Validation ») est égale à 0
 - 4.1 Exécuter à nouveau la requête et garder le résultat dans la variable RS1
 - 4.2 Initialiser la chaîne de caractères chaine2 à vide
 - 4.3 Tant que la variable RS1 contient un élément
 - 4.31 Affecter à la variable strnmgeo, le nom de la ville
 - 4.32 Affecter à la variable strnidgeo, le clé de la ville
 - 4.33 Initialiser la chaîne de caractères chaine à vide
 - 4.34 Si la taille du nom de la ville (strnmgeo) est inférieure ou égale à 20 caractères alors
 - Affecter à la variable chaine le nom de la ville (strnmgeo) et la cle (strnidgeo) pour remplir un élément de la liste déroulante

Afficher la variable chaine

Concaténer dans chaine2 la variable chaine

sinon

Prendre les 20 premiers caractères de la variable nom de la ville (strnmgeo) et affecter cela à la même variable strnmgeo. Affecter à la variable chaine le nom de la ville (strnmgeo) et la

cle (strnidgeo) pour remplir un élément de la liste déroulante

Afficher la variable chaine

Concaténer dans chaine2 la variable chaine

Fin_si

4.35 Avancer vers l'élément suivant (RS1.next)

Fin_tant

4.4 Fermer et vider la variable RS1

4.5 Affecter à la variable Session(« Villes »), la valeur chaine2, contenant toutes les villes (éléments) de la liste déroulante.

7.6 Affecter à la variable Session(« Validation »), la valeur 1 afin de savoir si nous avons déjà exécuté une fois la requête RS1.

Sinon

4.7 Afficher la valeur de Session(« Villes ») contenant tous les noms de villes dans la BD appartenant à un certain type de festivals.

Fin_si

Traduction en pseudo code

Programme ConstructionListeDeroulantContenantDesNomsDeVilles ;

Var

RS1 : enregistrement ; (*Pour stocker la réponse à la requête SQL*)

k : entier (*Compteur de villes*)

Début

(*Action 1*)

Connexion avec la BD.

(*Action 2*)

Si Session(« Validation »)==0 alors

 (*Action 2.1*)

```
SQL1= "SELECT DISTINCT indexgeo.nidgeo, thesaurusgeo.nmgeo
FROM thesaurusgeo INNER JOIN ((infofes INNER JOIN [index] ON
infofes.nidfes = index.nidfes) INNER JOIN indexgeo ON infofes.nidfes
= indexgeo.nidfes) ON thesaurusgeo.nidgeo = indexgeo.nidgeo
Where index.nidmotcle=10000 ORDER BY thesaurusgeo.nmgeo;"
```

 (*Action 2.2*)

```
Set RS1 = Server.CreateObject("ADODB.RecordSet")
```

```
RS1.Open sql1, OBJdbConnection, 3
```

 (*Action 2.3*)

```
k=0 ;
```

 (*Action 2.4*)

```
Tant que not eof (RS1) faire
```

```
        k=k+1 ;
```

```
        RS1.next
```

```
Fin_tant
```

 (*Action 2.5*)

```
RS1.close ;
```

```
RS1.nothing ;
```

 (*Action 2.6*)

```
Session(« nbrevilles »)=k ;
```

fin_si

(*Action 3*)

```
écrire('le nombre de villes indexées sur le site sont : ',
```

```
Session(« nbrevilles »)) ;
```

(*Action 4*)

Si Session(« Validation »)==0 alors

 (*Action 4.1*)

```

Set RS1 = Server.CreateObject("ADODB.RecordSet")
RS1.Open sql1, OBJdbConnection, 3
(*Action 4.2*)
chaine2= " " ;
(*Action 4.3*)
écrire('<form><input type=hidden name=req value=ville><select
name=ville>') ;
Tant que not eof RS1 faire
    (*Action 4.31*)
    strnmgeo=RS1(nmgeo) ;
    (*Action 4.32*)
    strnidgeo=RS1(nidgeo);
    (*Action 4.33*)
    chaine=" " ;
    (*Action 4.34*)
    Si taille(strnmgeo)<=20 alors
        chaine=strnmgeo
        écrire('<option value=strnidgeo>',chaine,'</option>') ;
        chaine2=chaine2 + "chaine "
    Sinon
        strnmgeo=(strnmgeo,20) ;
        chaine=strnmgeo + strnidgeo
        écrire('option value=strnidgeo>',chaine,'</option>') ;
        chaine2=chaine2 + "chaine "
    Fin_si
    (*Action 4.35*)
    RS1.next ;
Fin_tant ;
(*Action 4.4*)
RS1.close ;

```

```

RS1.nothing ;
(*Action 4.5*)
Session(« Villes »)=chaine2 ;
(*Action 4.6*)
Session (« Validation »)=1 ;
Sinon
(*Action 4.7*)
écrire (Session(« Villes ») ;
Fin_si ;
écrire('</select></form>') ;
Fin.

```

Implémentation

```

<p><!--Compteur des villes--> <!--et affichage des villes indexées pour la musique--> <!--
Set OBJdbConnection = Server.CreateObject("ADODB.Connection")
OBJdbConnection.Open "muser1"

if Session("Validation")=0 then

    SQL1= "SELECT DISTINCT indexgeo.nidgeo, thesaurusgeo.nmgeo FROM thesaurusgeo
    INNER JOIN ((infos INNER JOIN [index] ON infos.nidfes = index.nidfes) INNER JOIN
    indexgeo ON infos.nidfes = indexgeo.nidfes) ON thesaurusgeo.nidgeo = indexgeo.nidgeo
    Where index.nidmotcle=10000 ORDER BY thesaurusgeo.nmgeo;"

    Set RS1 = Server.CreateObject("ADODB.RecordSet")
    RS1.Open sql1, OBJdbConnection, 3

    Dim k
    k=0

    Do while Not RS1.EOF
    k=k+1
    RS1.MoveNext
    Loop
    RS1.close
    set RS1 = nothing

    Session("nbrevilles")=k

```

End if

```

> </p>
  <p><br>
    <font FACE="Verdana, Arial, Helvetica" SIZE="-2">Sélectionner parmi les <b><!--Cet if permet
d'afficher le nombre de villes fest. musique avec la valeur k pour la 1er. fois que l'utilisateur entre
dans cette page, dans la session en cours--> <%
    response.write Session("nbvilles")

</b>Villes
de musique indexées sur notre site.<br>
</p>
<form method="GET" name="rech_villes">

  <p><select
    onChange="window.open('../general/rqfes.asp?id1=10000&amp;type=musique&amp;p=../m
usique/default.asp&amp;p2=musiques&amp;req=ville&amp;ville=' +
    this.options[this.selectedIndex].value,'_top')"
    name="ville" size="1">
      <option selected value="default">Sélectionnez une ville</option>
</p>
<
if Session("Validation")=0 then

  Set RS1 = Server.CreateObject("ADODB.RecordSet")
  RS1.Open sql1, OBJdbConnection, 3

  Dim chaine2
  chaine2=""
  Dim chaine
  Do while Not RS1.EOF
  strnmgeo=RS1("nmgeo")
  strnidgeo=RS1("nidgeo")
  chaine=""

  if Len(strnmgeo)<=20 then
    chaine="<option value=" & strnidgeo & ">" & strnmgeo & "</option>"
    response.write chaine
    chaine2=chaine2 & chaine
  else
    strnmgeo=Left(strnmgeo,20)
    chaine="<option value=" & strnidgeo & ">" & strnmgeo & " ...</option>"
    response.write chaine

    chaine2=chaine2 & chaine
  End if

  RS1.MoveNext
Loop
RS1.close
set RS1 = nothing
OBJdbConnection.close
Session("Villes")=chaine2
Session("Validation")=1
>< else

```



```
<%=Session("Villes")%><%end if%> </select></font></p>  
</form>
```

ANNEXE B : Modification réalisée : Recherche par jour.**Actions**

1. Recevoir les paramètres (Emois et Ean) et affecter les variables mois et an avec ces valeurs.
 - mois=Emois
 - an=Ean
2. Ré-affectation des variables mois et an et initialisation de la variable journow.
 - 2.1. Si c'est la première fois que la page est visitée dans la session en cours (c'est-à-dire, session("Validation")=0) ou il n'y a pas de paramètres (mois= "" et an="") alors
 - Affecter à la variable mois, le mois du système
 - Affecter à la variable an, l'année du système
 - fin_si
 - 2.2. Si le mois est égal au mois du système et l'an est égal à l'année du système alors
 - Affecter à journow, le jour du système
 - Sinon
 - Affecter à journow la valeur de 1 (premier jour du mois)
 - Fin_si
3. Selon le mois actuel, affecter le nombre de jours du mois en cours à la variable joursmois et le nom du mois à la variable nomois
 - Choisir mois parmi :
 - Cas 1 : Affecter à joursmois le nombre de jours du mois en cours et affecter à la chaîne de caractères nomois le nom du mois.
 - Cas 2 : " "
 - .
 - .
 - .
 - Cas 12 : " "

4. Trouver la place du premier jour du mois dans la première ligne du calendrier et affecter cette valeur à la variable `inactifs` et `blanc`.

Affecter à `inactifs` la position du premier jour du mois, entre les 7 jours de la semaine (entre 1 et 7) - 1

Affecter à `blanc` la valeur de `inactifs`

5. Affichage d'entêtes du calendrier; le mois et les jours de la semaine.

6. Laisser `n` cellules vides (variable `blanc`) dans la première ligne du calendrier avant d'arriver à la place du 1er jour du mois

Tant que `inactifs` est supérieur à zéro et inférieur à 7

Ecrire une cellule vide du calendrier ("`<td></td>`")

Décrémenter en 1 la variable `inactifs`

7. Afficher en couleur grise les jours du mois qui sont déjà passés. De plus, chaque fois que le rang qu' occupe le jour est multiple de 7 (Mod 7) aller à la ligne (avec `</tr>`)

Initialiser le compteur `i` à 1

Tant que `i` est inférieur au jour d'aujourd'hui faire

Si je suis dans la septième colonne (limite du calendrier)

Ecrire la variable `i` (jour) dans une cellule et aller à la ligne

Sinon

Ecrire dans une cellule la variable `i` qui représente le jour

Incrémenter le compteur `i`

8. Faire des liens avec les jours du mois qui restent à partir d'aujourd'hui.

Tant que `i` est inférieur ou égal au nombre de jours de ce mois faire

Si je suis dans la septième colonne (limite du calendrier)

Faire un lien avec la variable `i` (jour) en envoyant comme des paramètres, à la page ASP réponse, le mois (`mois`), l'année (`an`) et le jour (`i`) et aller à la ligne du calendrier.

Sinon

Faire un lien avec la variable i (jour) en envoyant comme des paramètres, à la page ASP réponse, le mois (mois), l'année (an) et le jour (i).

Incrémenter la variable i.

9. Faire des liens vers le mois suivant et précédent en vérifiant quelques cas particuliers, c'est-à-dire, soit mois=12 ou mois=1.

si la date est la date du système (mois = au mois du système et an = à l'an du système) alors

si le mois est égal à 12 alors

Affecter au mois suivant (suimois) la valeur de 1

Affecter à l'année suivante (suian) la valeur de (an+1)

Faire un lien avec une flèche (dans un seul sens →), en envoyant comme paramètres, à cette même page ASP, le mois (suimois) et l'année (suian).

Sinon

Affecter au mois suivant (suimois) la valeur de (mois+1)

Affecter à l'an suivant (suian) la valeur de (an)

Faire un lien avec une flèche (dans un seul sens →), en envoyant comme paramètres, à cette même page ASP, le mois (suimois) et l'année (suian).

Fin_si

Sinon

Si le mois est égal à 12 alors

Affecter au mois suivant (suimois) la valeur de 1

Affecter au mois précédent (premois) la valeur de (mois -1)

Affecter à l'année suivante (suian) la valeur de (an + 1)

Affecter à l'année précédente (prean) la valeur de (an)

Faire un lien avec une flèche (← (mois précédent)), en envoyant comme paramètres, à cette même page ASP, le mois (premois) et l'année (prean).

Faire un lien avec une flèche ((mois suivant)→), en envoyant comme paramètres, à cette même page ASP, le mois (suimois) et l'année (suian).

Sinon

Si mois est égal à 1 alors

Affecter au mois suivant (suimois) la valeur de (mois + 1)

Affecter au mois précédent (premois) la valeur de 12

Affecter à l'année suivante (suian) la valeur de (an)

Affecter à l'année précédente (prean) la valeur de (an - 1)

Faire un lien avec une flèche (← (mois précédent)), en envoyant comme paramètres, à cette même page ASP, le mois (premois) et l'année (prean).

Faire un lien avec une flèche ((mois suivant)→), en envoyant comme paramètres, à cette même page ASP, le mois (suimois) et l'année (suian).

Sinon

Affecter au mois suivant (suimois) la valeur de (mois + 1)

Affecter au mois précédent (premois) la valeur de (mois - 1)

Faire un lien avec une flèche (← (mois précédent)), en envoyant comme paramètres, à cette même page ASP, le mois (premois) et l'année (an).

Faire un lien avec une flèche ((mois suivant)→), en envoyant comme paramètres, à cette même page ASP, le mois (suimois) et l'année (an).

Fin_si

Fin_si

Fin_si

Traduction en pseudocode

Programme Calendrier ;

Var

```

i : entier ; (*compteur*)
mois : chaîne de caractères ; (*variable utilisée pour garder le mois en cours*)
an : entier ; (*variable utilisée pour garder l'an en cours *)
journow : entier ; (*variable utilisée pour garder le jour d'aujourd'hui*)
nomois : chaîne de caractères ; (*variable utilisée pour garder le nom du mois
en cours*)
dat : date ;

```

Début

```

(*Action 1*)
lire (Emois) ;
lire(Ean) ;
mois=Emois ;
an=Ean ;
(*Action 2.1.*)
si Session("Validation")=0 or (mois="" and an="") alors
    mois = Month(Now) (*Mois du système*)
    an = Year(Now) (*An du système*)
fin_si
(*Action 2.2.*)
si ((mois=Month(Now)) and (an=Year(Now))) alors
    journow=Day(Now) (*Date d'aujourd'hui, jour*)
sinon
    journow=1
Fin_si

(*Action 3*)
choisir mois parmi :
cas 1 :
    joursmois=31 ;
    nomois= "Janvier" ;
cas 2 :
    Si (an Mod 4) = 0 alors (*si l'an est bissextile*)
        joursmois = 29
        nomois = "Février"
    sinon
        joursmois = 28
        nomois = "Février"
    Fin_si ;
cas 3 :
    joursmois=31 ;
    nomois="Mars" ;

```

cas 4 :
 joursmois=30 ;
 nomois="Avril" ;

cas 5 :
 joursmois=31 ;
 nomois="Mai" ;

cas 6 :
 joursmois=30 ;
 nomois="Juin" ;

cas 7 :
 joursmois=31 ;
 nomois= "Juillet" ;

cas 8 :
 joursmois=31 ;
 nomois="Août" ;

cas 9 :
 joursmois=30 ;
 nomois="Septembre" ;

cas 10 :
 joursmois=31 ;
 nomois= "Octobre" ;

cas 11 :
 joursmois=30 ;
 nomois= "Novembre" ;

cas 12 :
 joursmois=31 ;
 nomois="Décembre" ;

Fin_Cas

(*Action 4*)

dat =01 + " " + mois + " " + an ;
 inactifs=Weekday(dat)-1 ;
 blanc = inactifs ;

(*Action 5*)

écrire('<table><th>', nomois,'</th><tr><td>') ;
 écrire('<table><tr>') ;
 écrire('<td align=center>D</td>') ;
 écrire('<td align=center>L</td>') ;
 écrire('<td align=center>M</td>') ;
 écrire('<td align=center>M</td>') ;
 écrire('<td align=center>J</td>') ;
 écrire('<td align=center>V</td>') ;
 écrire('<td align=center>S</td></tr><tr>') ;
 (*Action 6*)

```

Tant que (inactifs > 0) AND (inactifs < 7) faire
  Ecrire('<td></td>');
  inactifs = inactifs - 1 ;
fin_tant ;
(*Action 7*)
i = 1
Tant que i < journow faire
  si ((i + blanc) Mod 7) = 0 alors
    écrire('<td align=center><font color=#999999>',i,'</td></tr><tr>');
  sinon
    écrire('<td align=center><font color=#999999>',i,'</td>');
  fin_si
  i = i + 1 ;
fin_tant ;
(*Action 8*)
Tant que i <= joursmois faire
  Si ((i + blanc) Mod 7) = 0 alors
    Ecrire('<td align=center><a href=../general/rqfes.asp?req=date
    & jour=', i, '& mois=', mois, '& annee=', an, '>', i
    , '</a></td></tr><tr>');
  Sinon
    écrire('<td align=center><a href=../general/rqfes.asp? req=date
    & jour=', i, '& mois=', mois, '& annee=', an, '>', i
    , '</font></a></td>');
  Fin_si ;
  i = i + 1 ;
Fin_tant ;
écrire('</tr>');
écrire('<tr><td colspan=2>');

(*Action 9*)
si ((mois=Month(now)) et (an=Year(now))) alors
  si mois=12 alors
    suimois=1 ;
    suian=an+1 ;
    écrire('</td><td colspan=2></td><td colspan=2></td><td
    colspan=2><a href=defaultcalend.asp?Emois=" & suimois & "&Ean="
    & suian & ">--></a></td>');
  sinon
    suimois=mois+1 ;
    suian=an ;
    écrire('</td><td colspan=2></td><td colspan=2></td><td
    colspan=2><a href=defaultcalend.asp?Emois=" & suimois & "&Ean="
    & suian & ">--></a></td>')

```



```

    Fin_si
sinon
    si mois=12 alors
        suimois=1 ;
        premois=mois-1 ;
        suian=an+1 ;
        prean=an ;
        écrire('<a href=defaultcalend.asp?Emois=" & premois & "&Ean=" &
prean & ">--</a></td><td colspan=2></td><td colspan=2></td><td
colspan=2><a href=defaultcalend.asp?Emois=" & suimois & "&Ean="
& suian & ">--></a></td>')
    sinon
        si mois=1 alors
            suimois=mois+1 ;
            premois=12 ;
            prean=an-1 ;
            suian=an ;
            écrire('<a href=defaultcalend.asp?Emois=" & premois &
"&Ean=" & prean & ">--</a></td><td colspan=2></td><td
colspan=2></td><td colspan=2><a href=defaultcalend.asp?
Emois=" & suimois & "&Ean=" & suian & ">--></a></td>')
        sinon
            suimois=mois+1 ;
            premois=mois-1 ;
            écrire('<a href=defaultcalend.asp?Emois=" & premois &
"&Ean=" & an & ">--</a></td><td colspan=2></td><td
colspan=2></td><td colspan=2><a href=defaultcalend.asp?
Emois=" & suimois & "&Ean=" & an & ">--></a></td>')
        Fin_si
    Fin_si
Fin_si
Fin_si
écrire('</td></tr></table>')
écrire('</td></tr></table>')
Fin.

```

Implémentation

```

<!--paramètres-->
< mois=Request.QueryString("Emois")%>
< an=Request.QueryString("Ean")%>

<!--code pour générer le calendrier--> <!--jour de la semaine--> < .
' La premiere fois que l'on entre dans la page, la variable
' session("Validation") est = à "0"

```

```

if Session("Validation")=0 or (mois="" and an="") then
    ' mois du système
    mois = Month(Now)
    ' an du système
    an = Year(Now)
end if

```

```

Amois=Month(Now)
Aan=Year(Now)
Dim m
Dim a
m="" & Amois
a="" & Aan
Dim resp1
Dim resp2
resp1=StrComp(mois,m)
resp2=StrComp(an,a)

```

```

if ((resp1=0) and (resp2=0)) then ' si c'est la date du système
    journow=Day(Now) ' le jour prend le jour d'aujourd'hui
else ' si c'est une autre date, initialiser la variable journow à 1
    journow=1
end if

```

> <!--Selon le mois actuel, le script affecte le nombre de jours à la variable joursmois et le nom du mois à la variable nomois--> <%

```

Select Case (mois)
    Case 1
        joursmois = 31
        nomois = "Janvier"
    Case 2
        If (an Mod 4) = 0 Then
            joursmois = 29
            nomois = "Février"
        Else
            joursmois = 28
            nomois = "Février"
        End If
    Case 3
        joursmois = 31
        nomois = "Mars"
    Case 4
        joursmois = 30
        nomois = "Avril"
    Case 5
        joursmois = 31
        nomois = "Mai"
    Case 6
        joursmois = 30
        nomois = "Juin"
    Case 7

```

```

    joursmois = 31
        nomois = "Juillet"
Case 8
    joursmois = 31
        nomois = "Août"
Case 9
    joursmois = 30
        nomois = "Septembre"
Case 10
    joursmois = 31
        nomois = "Octobre"
Case 11
    joursmois = 30
        nomois = "Novembre"
Case 12
    joursmois = 31
        nomois = "Décembre"
End Select

```

><!--cet équation permet de trouver le nombre de places vides avant d'afficher le premier jour du mois dans la première ligne du calendrier--><%

```

Dim dat
dat = 01 & " " & mois & " " & an
    inactifs = Weekday(dat) - 1
    blanc = inactifs

```

```

    > <!--Affichage d'entêtes du calendrier; le mois et les jours de la semaine--> <%
Response.write "<table><th><font FACE=Verdana, Arial, Helvetica SIZE=-2 color=#FF00AA>" &
    nomois & "</font></th><tr><td>"
Response.write "<table><tr>"
Response.write "<td align=center><font FACE=Verdana, Arial, Helvetica SIZE=-2
    color=#FF00AA>D</font></td>"
Response.write "<td align=center><font FACE=Verdana, Arial, Helvetica SIZE=-2
    color=#FF00AA>L</font></td>"
Response.write "<td align=center><font FACE=Verdana, Arial, Helvetica SIZE=-2
    color=#FF00AA>M</font></td>"
Response.write "<td align=center><font FACE=Verdana, Arial, Helvetica SIZE=-2
    color=#FF00AA>M</font></td>"
Response.write "<td align=center><font FACE=Verdana, Arial, Helvetica SIZE=-2
    color=#FF00AA>J</font></td>"
Response.write "<td align=center><font FACE=Verdana, Arial, Helvetica SIZE=-2
    color=#FF00AA>V</font></td>"
Response.write "<td align=center><font FACE=Verdana, Arial, Helvetica SIZE=-2
    color=#FF00AA>S</font></td></tr><tr>"
Response.Flush
    > <!--Le boucle suivant écrit des cellules vides dans la première ligne du calendrier avant d'arriver
à la place du 1er. jour du mois--> <%
Do While (inactifs > 0) AND (inactifs < 7)
    Response.write "<td></td>"
    inactifs = inactifs - 1
Loop

```

```

i = 1
  > <!--Ce boucle permet d'afficher en couleur gris les jous du mois qui sont déjà passés. De plus,
chaque place d'un jour qui est multiple de 7 (Mod 7)on va à la ligne avec </tr--> <
Do While i < journow
  If ((i + blanc) Mod 7) = 0 Then
    Response.write "<td align=center><font FACE=Verdana, Arial, Helvetica SIZE=-2
color=#999999>" & i & "</td></tr><tr>"
  Else
    Response.write "<td align=center><font FACE=Verdana, Arial, Helvetica SIZE=-2
color=#999999>" & i & "</td>"
  End If
  i = i + 1
Loop
  > <!--Ce boucle permet de faire des liens avec les jours du mois qui restent à partir d'aujourd'hui -
-> <

Do While i <= joursmois
  If ((i + blanc) Mod 7) = 0 Then
    Response.write "<td align=center><a
href=../general/rqfes.asp?id1=10000&amp;type=musique&amp;p=../musique/default.asp&amp;p2=
musiques&amp;req=date&amp;jour=" & i & "&amp;mois=" & mois & "&amp;annee=" & an & "><font
FACE=Verdana, Arial, Helvetica SIZE=-2 >" & i & "</font></a></td></tr><tr>"
  Else
    Response.write "<td align=center><a
href=../general/rqfes.asp?id1=10000&amp;type=musique&amp;p=../musique/default.asp&amp;p2=
musiques&amp;req=date&amp;jour=" & i & "&amp;mois=" & mois & "&amp;annee=" & an & "><font
FACE=Verdana, Arial, Helvetica SIZE=-2 >" & i & "</font></a></td>"
  End If
  i = i + 1
Loop

Response.write "</tr>"
Response.write "<tr><td colspan=2>"
if ((resp1=0) and (resp2=0)) then
  if mois=12 then
    suimois=1
    suian=an+1
    Response.write "</td><td colspan=2></td><td colspan=2> </td><td colspan=2><a
href=defaultcalend.asp?Emois=" & suimois & "&Ean=" & suian & ">--></a></td>"
  else
    suimois=mois+1
    suian=an
    Response.write "</td><td colspan=2></td><td colspan=2> </td><td colspan=2><a
href=defaultcalend.asp?Emois=" & suimois & "&Ean=" & suian & ">--></a></td>"
  end if
else
  if mois=12 then
    suimois=1
    premois=mois-1
    suian=an+1
    prean=an

```

```

Response.write "<a href=defaultcalend.asp?Emois=" & premois & "&Ean=" & prean
& "><--</a></td><td colspan=2></td><td colspan=2></td><td colspan=2><a
href=defaultcalend.asp? Emois=" & suimois & "&Ean=" & suian & ">--></a></td>"
else
if mois=1 then
    suimois=mois+1
    premois=12
    prean=an-1
    suian=an
    Response.write "<a href=defaultcalend.asp?Emois=" & premois & "&Ean="
& prean & "><--</a></td><td colspan=2></td><td colspan=2></td><td
colspan=2><a href=defaultcalend.asp?Emois=" & suimois & "&Ean=" &
suian & ">--></a></td>"
else
    suimois=mois+1
    premois=mois-1
    Response.write "<a href=defaultcalend.asp?Emois=" & premois & "&Ean="
& an & "><--</a></td><td colspan=2> </td><td colspan=2></td><td
colspan=2><a href= defaultcalend.asp?Emois=" & suimois & "&Ean=" & an
& ">--></a></td>"
end if
end if
end if
Response.write "</td></tr></table>"
Response.write "</td></tr></table>"
>

```

ANNEXE C : Une page réponse (thème, ville ou calendrier)

Fichier : rqfes.asp

```

< @ LANGUAGE="VBSCRIPT" %>
< Response.expires = 0 %>
< Response.Buffer = True %>
< strid1=Request.QueryString("id1")%>
< strtype=Request.QueryString("type")%>
< requete=Request.QueryString("req")%>
< numville=Request.QueryString("ville")%>
< strjour=Request.QueryString("jour")%>
< strmois=Request.QueryString("mois")%>
< stran=Request.QueryString("annee")%>
<
Set OBJdbConnection = Server.CreateObject("ADODB.Connection")
OBJdbConnection.Open "muser1"

>
<!--Script qui permet de faire la distinction entre requête par thème, par date ou par ville selon la
rubrique choisie par l'utilisateur -->
<
Select Case requete
  Case theme
    sql = " SELECT * FROM infofes INNER JOIN [index] ON infofes.nidfes = index.nidfes      WH
  Case date
    sql = " SELECT * FROM infofes INNER JOIN [index] ON infofes.nidfes = index.nidfes
    WHERE (index.nidmotcle = " & strid1 & " AND (" & strjour & " Between
    Day(infofes.dtedeb) And Day(infofes.dtefin)) AND ( " & strmois & " Between
    Month(infofes.dtedeb) And Month(infofes.dtefin)) And ( " & stran & " Between
    Year(infofes.dtedeb) And Year(infofes.dtefin))) ORDER BY infofes.moisnum,
    infofes.dtedeb;"
  Case ville
    sql = "SELECT * FROM (infofes INNER JOIN [index] ON infofes.nidfes = index.nidfes)
    INNER JOIN indexgeo ON infofes.nidfes = indexgeo.nidfes WHERE index.nidmotcle= "
    & strid1 & " AND indexgeo.nidgeo=" & numville & " ORDER BY infofes.moisnum,
    infofes.dtedeb;"
End Select

Set RS = Server.CreateObject("ADODB.RecordSet")
RS.Open sql, OBJdbConnection, 3
RS.PageSize = 10

>
<html>

<head>...

< !—Affichage des résultats - ->

```

```

<td><table>
  <tr>
    <td>
      <a href="#">=<%= RS("nmfes")%></font></a></td>
    </tr>
    <tr>
      <td><font FACE="Arial, Helvetica" SIZE="-2"><%= RS("vil")%>, <%= RS("dep")%>,
        <%= RS("reg")%>, <%= RS("mois")%><br>
        </font></td>
    </tr>
  </table>
</td>
< :
  RS.MoveNext
  Loop
>
  set RS = RS.NextRecordSet
  Loop

  OBJdbConnection.Close
  set rs = nothing
  set OBJdbConnection = nothing

...</body>
</html>

```

ANNEXE D : Annuaire de villes de France

Liste de URL's, contenant des informations concernant les villes de France, répertoriés avec le logiciel PowerMarks :

http://alsace-info.com/francais/tourisme/a-visiter/
http://www.yahoo.fr/Exploration_geographique/Pays/France/Villes/
http://www.communes.com/
http://perso.wanadoo.fr/entreprise/
http://www.annu-ville.com/
http://www.frenchflavor.com
http://www.enprovence.com/tableau_villes.html
http://www.provenceweb.fr/f/var/villages.htm
http://www.najac.com/
http://www.environnement.gouv.fr/ACTUA/ops/Sansvoiture/listevilles.htm#s
http://www.ntmm-gerber.fr/ntmm03.html
http://www.savoie-tourisme.com/v4/index.htm
http://www.histo.com
http://www.webring.org/cgi-bin/webring?ring=v2f;list
http://www.vpah.culture.fr/vpah/sommaire/sommaire.htm
http://arch.ipt.univ-paris8.fr/vdm/noticeshistorique/noticeshist.html
http://perso.wanadoo.fr/jacques.managau/
http://www.gaf.tm.fr/index/villes.htm

ANNEXE E : Rubrique des villes : Recherche par ville.**Actions**

1. Se connecter avec la BD ACCESS
2. Affecter au compteur de villes(k) la valeur zéro.
3. Requête SQL pour trouver les villes (nom) qui ont au moins un site internet dans la BD
 Exécuter la requête SQL et garder le résultat dans la variable RS1.
4. Tant que ne soit pas le dernier élément de RS1
 Incrémenter le compteur k en 1.
 Aller au suivant élément de RS1
5. Afficher la valeur de k.
6. Afficher les villes
 - 6.1 Requête SQL pour trouver les villes (nom et clef) qui ont au moins un site Internet dans la BD
 Exécuter la requête SQL et garder le résultat dans la variable RS2.
 - 6.2
 Tant que ne soit pas le dernier élément de RS2
 - 6.2.1
 Si la taille du RS2(nomville) est inférieure au 20 caractères alors
 Afficher l'élément RS2(nomville) pour remplir la liste
 déroulante
 - sinon
 Affecter la variable ville avec les premiers 20 caractères de la variable RS2(nomville)
 Afficher l'élément RS2(nomville) pour remplir la liste
 déroulante
 - fin si ;
- Afficher l'élément RS2(clefville) pour le mettre à côté de chaque ville
 Aller à l'élément suivant de RS2
- fin tant.

Traduction en pseudocode

Programme Nombre_Villes ;

Var

 K : entier ; (*Compteur de villes*)

 RS1 : liste d'enregistrements ;

 RS2 : liste d'enregistrements ;

Début

 (* Action 1*)

 Connexion à la BD ;

 (* Action 2*)

 K=0 ;

 (* Action 3*)

 SQL1= "SELECT DISTINCT thesaurusgeo.nmgeo FROM thesaurusgeo
 INNER JOIN Urlvillestest ON thesaurusgeo.nidgeo = Urlvillestest.nidgeo;"

 Set RS1 = Server.CreateObject("ADODB.RecordSet")

 RS1.Open sql1, OBJdbConnection, 3

 (* Action 4*)

 Tant que not eof RS1 faire

 k = k + 1 ;

 RS1.Next

 fin tant ;

 (* Action 5*)

 écrire('Le nombre de villes est: ' k) ;

 (* Action 6*)

 (*Action 6.1*)

 SQL2= "SELECT DISTINCT thesaurusgeo.nmgeo, thesaurusgeo.nidgeo
 FROM thesaurusgeo INNER JOIN Urlvillestest ON thesaurusgeo.nidgeo =
 Urlvillestest.nidgeo;"

 Set RS2 = Server.CreateObject("ADODB.RecordSet")

 RS2.Open sql2, OBJdbConnection, 3

 (*Action 6.2*)

 Tant que not eof RS2 faire

 (*Action 6.2.1*)

 Si taillechaîne(RS2(nomville))<=20 alors

```

        écrire('Le nom de ville : ' RS2(nomville)) ;
    sinon
        ville=chaine(RS2(nomville),20) ;
        écrire('Le nom de ville : ' ville) ;
    fin si ;
    écrire('La clef de la ville est: ' RS2(clefville)) ;
    RS2.Next
fin tant ;

```

Fin.

Implémentation :

```
<!--Connexion avec la BD ACCESS-->
```

```
<
```

```
Set OBJdbConnection = Server.CreateObject("ADODB.Connection")
```

```
OBJdbConnection.Open "muser1"
```

' Requête SQL (Cette requête doit être formulée avec SQL ACCESS), nous utilisons ' le mot DISTINCT pour éviter d'avoir des doublons.

```
SQL1= "SELECT DISTINCT thesaurusgeo.nmgeo FROM thesaurusgeo INNER JOIN Urvillestest
ON thesaurusgeo.nidgeo = Urvillestest.nidgeo;"
```

' Affecter le résultat de la requête à la variable RS1

```
Set RS1 = Server.CreateObject("ADODB.RecordSet")
```

```
RS1.Open sql1, OBJdbConnection, 3
```

' Initialisation à zéro du compteur (k) des villes qui ont un site internet

```
Dim k
```

```
k=0
```

' Compter le nombre de villes qui ont au moins un site internet

```
Do while Not RS1.EOF
```

```
k=k+1
```

```
RS1.MoveNext
```

Loop

' Fermer et vider la variable utilisée (RS1) pour garder le résultat de la requête

```
RS1.close
```

```
set RS1 = nothing
```

```
>
```

```
<font FACE="Verdana, Arial, Helvetica" SIZE="-2"><p>Sélectionner parmi les <b>
```

```
<!--Affichage du nombre de Villes (Compteur)-->
```

```
< =k >
```

```
</b>Villes indexées qui ont au moins un site internet.<br> </p>
```

```
<!--Affichage des noms de villes à l'aide d'un comboBox-->
```

```
<form method="GET" name="rech_url_villes">
```

```
<p><select
```

```
onChange="window.open('../general/rqville.asp?ville=' +
```

```
this.options[this.selectedIndex].value, '_top')"
```

```
name="ville" size="1">
```

```
<option selected value="default">Sélectionnez une ville</option>
```

```
<
```

' Requête SQL (Cette requête doit être formulée avec SQL ACCESS), nous utilisons ' le mot DISTINCT pour éviter d'avoir des doublons.

```
SQL2= "SELECT DISTINCT thesaurusgeo.nmgeo, thesaurusgeo.nidgeo FROM thesaurusgeo
INNER JOIN Urlvillestest ON thesaurusgeo.nidgeo = Urlvillestest.nidgeo;"
```

' Affecter le résultat de la requête à la variable RS2

```
Set RS2 = Server.CreateObject("ADODB.RecordSet")
```

```
RS2.Open sql2, OBJdbConnection, 3
```

' Afficher les noms des villes avec la clef

```
Do while Not RS2.EOF
```

```
strnmgeo=RS2("nmgeo")
```

```
strnidgeo=RS2("nidgeo")
```

'Vérifier la taille du nom de la ville

```
if Len(strnmgeo)<=20 then
    response.write "<option value=" & strnidgeo & ">" & strnmgeo & "</option>"
else
    strnmgeo=Left(strnmgeo,20)
    response.write "<option value=" & strnidgeo & ">" & strnmgeo & " ...</option>"
End if
```

```
RS2.MoveNext
```

```
Loop
```

' Fermer et vider la variable utilisée (RS2) pour garder le résultat de la requête

```
RS2.close
```

```
set RS2 = nothing
```

' Fermer la connexion avec la BD

```
ObjdbConnection.close
```

```
> </select></font></p>
```

```
</form>
```

ANNEXE F : Une page réponse, recherche par ville

Actions

1. Prendre le paramètre envoyé par l'utilisateur (numéro de la ville) au moment où il fait son choix
2. Faire la connexion avec la BD ACCESS.
3. Initialisation de l'indice i à zéro
4. Faire la requête SQL pour trouver toutes les caractéristiques de la ville sélectionnée (nom, adresse URL, label etc...)
Exécuter la requête et garder le résultat dans la variable RS1
5. Afficher les résultats de la dernière requête
 - Tant que ne soit pas le dernier élément de RS1
 - Afficher les caractéristiques de la ville choisie
 - Aller au suivant élément de RS1
 - fin tant.

Traduction en pseudocode

Programme Caracteristiques_Villes ;

Var

 Numville : chaîne de caractères ; (*variable utilisée pour prendre le paramètre contenant le numéro de la ville choisie par l'utilisateur*)

 RS1 : liste d'enregistrements ;

Début

 (* Action 1*)

 écrire('Saisir le numéro de la ville désirée : ');

 lire(numville) ;

 (* Action 2*)

 Connexion à la BD ;

 (* Action 3*)

 i=1 ;

 (* Action 4*)

 sql = " SELECT Urlvillestest.ville, Urlvillestest.urlville, Urlvillestest.motcles, Urlvillestest.nidgeo, Urlvillestest.langue, thesaurusgeo.label , Urlvillestest.email, Urlvillestest.label FROM thesaurusgeo INNER JOIN

```

Urlvillestest ON thesaurusgeo.nidgeo = Urlvillestest.nidgeo WHERE
(Urlvillestest.nidgeo)=" & numville & ";"
Set RS1 = Server.CreateObject("ADODB.RecordSet")
RS1.Open sql1, OBJdbConnection, 3
(* Action 5*)
Tant que not eof RS1 faire
    écrire('Le caractéristiques de la ville sont : ' Carac_Ville[i]) ;
    RS.Next
fin tant ;

```

Fin.

Implémentation

Fichier : rqville.asp

```

< @ LANGUAGE="VBSCRIPT" %>
<... Response.Buffer = True %>
< numville=Request.QueryString("ville")%>
<html>
<head>
...
<
Set OBJdbConnection = Server.CreateObject("ADODB.Connection")
OBJdbConnection.Open "muser1"
>
<!--Script qui permet de faire la requête par ville et afficher les résultats--> <:

sql = " SELECT Urlvillestest.ville, Urlvillestest.urlville, Urlvillestest.motcles, Urlvillestest.nidgeo,
Urlvillestest.langue, thesaurusgeo.label, Urlvillestest.email, Urlvillestest.label FROM thesaurusgeo
INNER JOIN Urlvillestest ON thesaurusgeo.nidgeo = Urlvillestest.nidgeo WHERE
(Urlvillestest.nidgeo)=" & numville & ";"

Set RS = Server.CreateObject("ADODB.RecordSet")
RS.Open sql, OBJdbConnection, 3
RS.PageSize = 10

> </p>
<!-- Affichage de résultats -->

< response.write "Vous cherchez la ville de&nbsp;" : > < = RS("ville") : >
<p> Résultat(s) pour <% = RS("ville")%>

```

```
< Do While Not RS.EOF
```

```
<p>< % = RS("ville")%>
```

```
<a href="< % = RS("urlville")%>"><% = RS("urlville")%></a>, <% = RS("label")%>, <% = RS("email")%><br>
```

```
<
```

```
    RS.MoveNext
```

```
    Loop
```

```
>
```

```
<
```

```
    OBJdbConnection.Close
```

```
    set rs = nothing
```

```
    set OBJdbConnection = nothing
```

```
>
```

```
...
```

```
</body>
```

```
</html>
```


ANNEXE G : Une page réponse, recherche par thème

Actions

Une fois le thème choisi par l'utilisateur, il fallait créer une autre page pour répondre à ses besoins.

1. Prendre le paramètre : numéro du thème choisi (clef)
2. Connexion avec la BD
3. Requête SQL ; pour trouver les villes qui traitent du thème choisi
Exécuter la requête et garder le résultat dans RS1
4. Afficher les résultats de la dernière requête
Tant que ne soit pas le dernier élément de la RS1
Afficher les caractéristiques des villes contenues dans RS1
Aller à l'élément suivant de RS1
fin tant.

Traduction en pseudocode

Programme Caractéristiques_Villes ;

Var

clef : chaîne de caractères ; (*variable utilisée pour prendre le paramètre contenant le numéro du thème choisi par l'utilisateur*)

RS1 : liste d'enregistrements ;

Début

(* Action 1*)

écrire('Saisir le thème désirée : ');

lire(clef) ;

(* Action 2*)

Connexion à la BD ;

(* Action 3*)

```
SQL1 = " SELECT  Urlvillestest.ville,      Urlvillestest.urlville      ,
Urlvillestest.motcles,      Urlvillestest.langue,      Urlvillestest.email,
thesaurusgeo.label, thesaurus.motcle FROM (thesaurusgeo INNER JOIN
Urlvillestest ON thesaurusgeo.nidgeo = Urlvillestest.nidgeo) INNER JOIN
```

```

indexvilles ON Urlvillestest.numero = indexvilles.numero WHERE
indexvilles.nidmotcle=" & clef & ";"
Set RS1 = Server.CreateObject("ADODB.RecordSet")

RS1.Open sql1, OBJdbConnection, 3

(* Action 4*)
Tant que not eof RS1 faire
    écrire('Les caractéristiques de la ville sont : ' RS 1(ville), RS1(urlville),
    RS1(langue), RS1(email)) ;
    RS1.Next ;
fin tant ;
Fin.

```

Implémentation

rqville.asp

```

< @ LANGUAGE="VBSCRIPT" %>
< Response.expires = 0 %>
< Response.Buffer = True %>
< clef=Request.QueryString("mots")%>
<html>
...

<p><
Set OBJdbConnection = Server.CreateObject("ADODB.Connection")
OBJdbConnection.Open "muser1"
>
<!--Script qui permet de faire la requête par theme et afficher les résultats --> <

sql = " SELECT Urlvillestest.ville, Urlvillestest.urlville, Urlvillestest.motcles, Urlvillestest.langue,
Urlvillestest.email, thesaurusgeo.label, thesaurus.motcle FROM (thesaurusgeo INNER JOIN
Urlvillestest ON thesaurusgeo.nidgeo = Urlvillestest.nidgeo) INNER JOIN indexvilles ON
Urlvillestest.numero = indexvilles.numero WHERE indexvilles.nidmotcle=" & clef & ";"

Set RS = Server.CreateObject("ADODB.RecordSet")
RS.Open sql, OBJdbConnection, 3
> </p>
<!-- Affichage des résultats -->

response.write "Villes par theme: "%><%=RS("motcle") >

Résultat(s) pour <%=RS("motcle")%>

< Do While Not RS.EOF
>

```

```
< = RS("ville")%><a href="<%=RS("urlville")%>"><%= RS("urlville")%></a>, <%= RS("label")%>,
< = RS("email")%>
< .
  RS.MoveNext
  Loop
>
  OBJdbConnection.Close
  set RS = nothing
  set OBJdbConnection = nothing
.>

...
</html>
```

ANNEXE H : Une page réponse, villes d'«Art et d'Histoire»

Actions

Une fois le thème « Villes d'Art et d'Histoire » choisi, il fallait créer une autre page pour afficher les villes correspondant à ce choix.

1. Prendre le paramètre : label
2. Connexion avec la BD
3. Requête SQL ; pour trouver les villes d' « Art et d'Histoire »
Exécuter la requête et garder le résultat dans RS1
4. Afficher les résultats de la dernière requête
 - Tant que ne soit pas le dernier élément de la RS1
 - Afficher les caractéristiques des villes contenues dans RS1
 - Aller à l' élément suivant de RS1
 - fin tant.

Traduction en pseudocode

Programme Caractéristiques_Villes ;

Var

strlabel : chaîne de caractères ; (*variable utilisée pour prendre le paramètre contenant le label de villes choisi*)

RS1 : liste d'enregistrements ;

Début

(* Action 1*)

écrire('Saisir le label : ') ;

lire(strlabel) ;

(* Action 2*)

Connexion à la BD ;

(* Action 3*)

```

SQL1 = " SELECT Urlvillestest.ville, Urlvillestest.urlville, Urlvillestest.langue,
Urlvillestest.email, thesaurusgeo.label FROM thesaurusgeo INNER JOIN
Urlvillestest ON thesaurusgeo.nidgeo = Urlvillestest.nidgeo WHERE
thesaurusgeo.label LIKE '%" & strlabel & "%%' ;"
Set RS1 = Server.CreateObject("ADODB.RecordSet")
RS1.Open sql1, OBJdbConnection, 3
(* Action 4*)
Tant que not eof RS1 faire
écrire('Les caractéristiques de la ville sont : ' RS 1(ville), RS1(urlville),
RS1(langue), RS1(email),RS1(label)) ;
        RS1.Next ;
fin tant ;

```

Fin.

Implémentation

rqville.asp

```

< @ LANGUAGE="VBSCRIPT" %>
< Response.expires = 0 %>
< Response.Buffer = True %>
< strlabel=Request.QueryString("label")%>

<html>
...
<p><%
Set OBJdbConnection = Server.CreateObject("ADODB.Connection")
OBJdbConnection.Open "muser1"
>
<!--Script qui permet de faire la requête par « label » et affiche les résultats -->
<
sql = " SELECT Urlvillestest.ville, Urlvillestest.urlville, Urlvillestest.langue, Urlvillestest.email,
thesaurusgeo.label FROM thesaurusgeo INNER JOIN Urlvillestest ON thesaurusgeo.nidgeo =
Urlvillestest.nidgeo WHERE thesaurusgeo.label LIKE '%" & strlabel & "%%' ;"

Set RS = Server.CreateObject("ADODB.RecordSet")
RS.Open sql, OBJdbConnection, 3
>

<!--Affichage des résultats -->

<p><% if flaglabel then
response.write "Villes avec le label: " %><%=RS("label")%><%

```

```

response.write "(Villes d'Art et d'Histoire)"%>

<p>Résultat(s) pour <b><%if flaglabel then%><%=RS("label")%><%
response.write "(Villes d'Art et d'Histoire)"%>
<
  Do While Not RS.EOF
    ><%= RS("ville")%> <a href="<%=RS("urlville")%>"><%= RS("urlville")%></a>, <%=
RS("label")%>, <%= RS("email")%><br>
<%=
    RS.MoveNext
  Loop
<
  OBJdbConnection.Close
  set RS = nothing
  set OBJdbConnection = nothing

  >
...
</html>

```

ANNEXE I : Fichiers utilisés avec le logiciel NetMaps.

Fichier testfrance.html

```

<html>

<head>
<title>NetMaps: France Map</title>
</head>

<body bgcolor="EEEEEE">

<h1 align="center">Carte de France</h1>

<p align="center">
<applet code="NetMaps2.class" archive="hash_classes.zip" width="550" height="400">
  <param name="ConfigFile" value="testfrance.cfg">
  <param name="showLegend" value="true">
</applet>
</p>

<p align="center">File format: MapInfo MIF/MID<br>
Size: approx. 1MB </p>
</body>
</html>

```

Fichier testfrance.cfg

```

NetMapsConfig 20
BeginGlobals
  NumLayers = 2
EndGlobals
#
BeginLayer
  Title = Departements
  GeoFile = F_dept.mif MIF
  DataFile = F_dept.mid MID
#
  SetPen = (1,2,#aaffaa)
  SetBrush = (2,#aaffaa,#000000)
EndLayer
#
BeginLayer
  Title = Villes
  GeoFile = F_villes.mif MIF
  DataFile = F_villes.mid MID
  SetSymbol (34,#0000ff,5)
#
  SetPen = (1,2,#0000ff)
  SetBrush = (2,#0000ff,#ff0000)
EndLayer
#

```

Fichier f_villes.mif

Version 2

Delimiter ","

Index 1,2

CoordSys Earth Projection 3, 9999, 6, -168, -60, 320, 0, 0, 0, 0, 0, "m", 2.337229104484, 46.8,
45.90287723937, 47.69712276063, 600000, 2200000 Bounds (-113976741.175, -106377291.405)
(115176741.175, 122776190.945)

Columns 3

CODE Char(5)

NOM Char(50)

POP90 Decimal(15, 3)

Data

Point 833400.3 2110600.1
Symbol (34,16711680,5)
Point 869200.2 2129800.3
Symbol (34,16711680,5)
Point 860700.4 2089900.1
Symbol (34,16711680,5)
Point 822900.2 2138000.1
Symbol (34,16711680,5)
Point 894200.2 2159800.3
Symbol (34,16711680,5)
Point 890800.2 2146400.2
Symbol (34,16711680,5)
Point 886300.2 2154900.3
Symbol (34,16711680,5)

...

Fichier f_villes.MID

"01004","Amberieu-en-Bugey",10458.000
"01033","Bellegarde-sur-Valserine",11153.000
"01034","Belley",7820.000
"01053","Bourg-en-Bresse",40976.000
"01143","Divonne-les-Bains",5588.000
"01160","Fereny-Voltaire",6406.000
"01173","Gex",6616.000
"01202","Lagnieu",5688.000
"01244","Meximieux",6231.000
"01249","Miribel",7684.000
"01262","Montluel",5968.000
"01283","Oyonnax",23875.000
"01289","Peronnas",5353.000
"01354","Saint-Genis-Pouilly",5707.000
"01427","Trevoux",6090.000
"02095","Bohain-en-Vermandois",6955.000

...

Fichier f_dept.mif

Version 300

Charset "WindowsLatin1"

Delimiter ","

Index 1

CoordSys Earth Projection 3, 1002, "m", 0, 46.8, 45.898918964419, 47.696014502038, 600000, 2200000 Bounds (-113967455.417, -106367759.649) (115167455.417, 122767151.185)

Columns 4

CODE_DEPART Char(5)

NOM Char(50)

POPSDC Integer

SUPERFICIE_KM2 Decimal(20, 2)

Data

Region 1

388

862045.3 2084240.8

861842.5 2083727.5

862160.4 2083138.9

862855.7 2081850.8

862050.4 2080522.9

860732.2 2079424.1

860867.3 2078156.8

860948.3 2076881.5

860791.7 2076747

858355.3 2075341.1

857470 2073566.2

856353 2073756.7

855875.8 2074309.2

855960.8 2074599.6

854987.2 2075706.1

854619.9 2078022.7

853400.4 2079188.4

852323.8 2079804.9

851270.3 2080143.6

850601.4 2079976.2

850606.5 2080774

850023.7 2081893.6

...

Fichier f_dept.mid

"01", "AIN", 471019, 5772.23

"02", "AISNE", 537259, 7403.13

"2A", "CORSE-DU-SUD", 118808, 4119.79

"2B", "HAUTE-CORSE", 131563, 4695.99

"03", "ALLIER", 357710, 7405.92

"04", "ALPES-DE-HAUTE-PROVENCE", 130883, 6941.79

"05", "HAUTES-ALPES", 113300, 5691.78

"06", "ALPES-MARITIMES", 971829, 4294.10

"07", "ARDECHE", 277581, 5583.94

"08", "ARDENNES", 296357, 5197.16

"09","ARIEGE",136455,4893.00
"10","AUBE",289207,6006.94
"11","AUDE",298712,6369.99
"12","AVEYRON",270141,8672.31
"13","BOUCHES-DU-RHONE",1759371,5077.38
"14","CALVADOS",617131,5595.37
"15","CANTAL",158723,5769.28
...

ANNEXE J : Code utilisé pour la conception des servlets.

Servlet

ParamRequete.java

```

import java.io.*;
import java.sql.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ParamRequete extends HttpServlet
{
    ResourceBundle rb = ResourceBundle.getBundle("LocalStrings");
    private Connection conn;

    public void init(ServletConfig conf) throws ServletException
    {
        super.init(conf);
        try {
/*Connexion avec la Base de Données*/
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            Conn = DriverManager.getConnection("jdbc:odbc:conntest1");
        }
        catch (Exception e)
        {
            throw(new UnavailableException(this, "Désolé, La connexion avec la Base de
            Données n'a pas été réussi!"));
        }
    }

    public void doGet(HttpServletRequest request,HttpServletResponse response)
    throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        try
        {
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Carte Villes</title>");
            out.println("</head>");
            out.println("<body bgcolor=\"white\">");
            out.println("<h2>Repr&eacute;sentation cartographique</h2>");

/* affecter à la variable villeName la valeur du paramètre name qui contient le nom de la ville
            */

```

```

String villeName = request.getParameter("name");

/*Faire une requête SQL qui donne comme résultat les caractéristiques de la ville choisie
( ?), le nombre de figures qui la composent et leur respectives coordonnées*/

PreparedStatement pstmt = conn.prepareStatement("select villes.pop90, villes.code,
villes.nbrefigures, figure.typefig, figure.nbrePoints, Coord.x, Coord.y from
figure,Coord,villes where ((Coord.clefigure = figure.cletable) AND (villes.code =
figure.codeVille) AND (villes.nom like ? ))");

/*donner la valeur à chaque paramètre dans la requête SQL (?)
dans ce cas il y en a seulement un */

pstmt.setString(1,villeName);
/*Exécuter la requête*/
ResultSet selectresult = pstmt.executeQuery();

/*Se positionner dans le premier élément enregistrement du résultat de la requête */

boolean bool = selectresult.next();

out.println("<P>");
out.println("<applet code=\"Hola2.class\"
codebase=\"http://localhost:8080/examples/servlets/\" width=535 height=475>");

/*Envoyer à l'applet le nom d'une image (France.jpg) pour l'afficher en arrière plan */

out.println("<param name=\"img\"
value=\"http://localhost:8080/examples/servlets/France.jpg\">");

/*Envoyer le nom de la ville à l'applet Dessin.class*/

out.println("<param name=ville value=" + villeName+ ">");

int npoints=0;
int i=1;
boolean flag=true;
int nbrefig=0;
String tfig="";
String pop="";

/*Lire les résultats et les envoyer comme paramètres vers l'applet Dessin.class*/

while (bool)
{
out.println("<param name=x" + i +" value=" + ((selectresult.getInt(6))/10000) + ">");
out.println("<param name=y" + i +" value=" + ((selectresult.getInt(7))/10000) + ">");
if (flag)
{
npoints=selectresult.getInt(5);
tfig=selectresult.getString(4);
nbrefig=selectresult.getInt(3);
pop=selectresult.getString(1);
flag=false;
}
}

```

```

        }
        i++;
        bool = selectresult.next();
    }
    out.println("<param name=points value="+ npoints +">");
    out.println("<param name=type value="+ tfig +">");
    out.println("<param name=nfig value="+ nbrefig +">");
    out.println("<param name=popu value="+ pop +">");
    out.println("</applet>");
    out.println("</p>");
    out.println("</body>");
    out.println("</html>");
}

catch (SQLException sqle)
{
    System.out.println(sqle);
}
}

public void destroy()
{
    try
    {
        conn.close();
    }
    catch(SQLException sqle)
    {
        System.out.println(sqle);
    }
}

public void doPost(HttpServletRequest request,
                    HttpServletResponse response)
    throws IOException, ServletException
{
    doGet(request, response);
}
}

```

Applet

Dessin.java

```

/*
 * <applet code="Hola2" width=600 height=400>
 * <param name="img" value="France.jpg">
 * <param name=ville value=Lyon>
 * <param name=points value=5>
 * <param name=type value=poly>
 * <param name=nfig value=1>

```

```

* <param name=popu value=50000>
* <param name=x1 value=50>
* <param name=x2 value=120>
* <param name=x3 value=75>
* <param name=x4 value=20>
* <param name=x5 value=50>
* <param name=y1 value=5>
* <param name=y2 value=80>
* <param name=y3 value=110>
* <param name=y4 value=80>
* <param name=y5 value=10>
* </applet>
*/

import java.awt.*;
import java.util.*;
import java.io.*;
import java.applet.*;

public class Dessin extends Applet
{
    Image art;
    public void init()
    {
        /*Affecter la variable art avec le nom du fichier France.jpg*/
        art=getImage(getDocumentBase(),getParameter("img"));
    }

    public void paint(Graphics g)
    {
        /*Afficher l'image de la France (format jpg) en arriere plan. */
        g.drawImage(art,0,0,this);

        /*Prendre tous les paramètres envoyés par le servlet*/
        String nomville = getParameter("ville");
        int N = Integer.parseInt(getParameter("points"));
        int nfigu = Integer.parseInt(getParameter("nfig"));
        String typef = getParameter("type");
        String popul = getParameter("popu");
        int x1;
        int y1;
        int x[];
        int y[];
        x=new int[N];
        y=new int[N];
        String c1= "point" ;
        String c2= "poly" ;
        String chainex;
        String chainey;
        for (int i=0;i<N;i++)

```

```

{
    chainex="x" + (i+1);
    chainey="y" + (i+1);
    x[i]=Integer.parseInt(getParameter(chainex));
    y[i]=Integer.parseInt(getParameter(chainey));
}

```

/*Afficher les données envoyées par le servlet*/

```

g.setColor(Color.red);
g.drawString("Ville: " + nomville,50,50);
g.setColor(Color.black);
g.drawString("Population: " + popul,50,60);

```

/*dessiner les figures avec les coordonnées envoyées par le servlet*/

```

g.setColor(Color.blue);
if c1.equals(type) /*type point*/
{
    x1=x[0];
    y1=y[0];
    g.drawOval(x1,y1,10,10);
    g.fillOval(x1,y1,10,10);
}
else
{
    if c2.equals(type) /*type polygone*/
    {
        g.drawPolygon(x,y,N);
        g.fillPolygon(x,y,N);
    }
}
}
}
}

```

Page ASP

Afficher les villes qui se trouvent dans la BD dans une liste déroulante et appeler le servlet (ParamRequete) à partir du formulaire.

DefaultVilles.asp

```

<% @ LANGUAGE="VBSCRIPT" %>
<% Response.expires = 0 %>
<% Response.Buffer = True %>
<html>

```

```

:
:

```

```

.<%Set OBJdbConnection = Server.CreateObject("ADODB.Connection")
OBJdbConnection.Open "conntest1"

```

'Requête qui trouve tous les noms de villes dans la table Villes, sans doublons

```
SQL3= "SELECT DISTINCT villes.nom FROM villes ORDER BY villes.nom;"
```

```
Set RS3 = Server.CreateObject("ADODB.RecordSet")
RS3.Open sql3, OBJdbConnection, 3%> </p>

```

```
<p><b><font color="#8800FF">Choisir une ville</font></b> </p>

```

***' Construction du Formulaire qui contient les nom de villes et appelle le servlet
«ParamRequete»***

```

<form action="http://localhost:8080/examples/servlet/ParamRequete" method="POST">
  <p>Ville: <select name="firstname" size="1">
<%
Do while Not RS3.EOF
  strnmgeo=RS3("nom")
  chaine=""

  if Len(strnmgeo)<=20 then
    chaine="<option value=" & strnmgeo & ">" & strnmgeo & "</option>"
    response.write chaine
  else
    strnmgeo=Left(strnmgeo,20)
    chaine="<option value=" & strnmgeo & ">" & strnmgeo & " ...</option>"
    response.write chaine
  End if

RS3.MoveNext
Loop
RS3.close
set RS3 = nothing
OBJdbConnection.close%>      </select><br>
  <input type="submit"> </p>
</form>
.
.
.
</html>

```


Bibliographie

▪ *Monographies :*

A.D.D.E. (Analyse des données pour la décision dans l'entreprise). *MapInfo Professional 5.0, Guide de l'utilisateur MapInfo Corporation.* Levallois Perret Cedex : A.D.D.E. 1992-1994.

A.D.D.E. (Analyse des données pour la décision dans l'entreprise). *MapInfo Professional 5.0, Manuel de Référence MapInfo Corporation.* Levallois Perret Cedex : A.D.D.E. 1992-1994.

BARBEY Adélaïde (Direction). *Guides Voir.* Adaptation, révision et mise à jour par FAUVEL Jean-Jacques, FURET Pierre-Jean, GÉDOUIN Hélène, et al. Première édition. Paris, France : Hachette Littérature Générale, 1995. 672 p. Guides Voir, France. ISBN : 2-01-242106-7.

BROPHY Keith, KOETS Timothy. *Le programmeur VBScript.* Paris : Simon&Schuster Macmillan, 1996. ISBN 2-7440-0184-8.

CAMPBELL Bruce, DARNELL Rick. *Le Programmeur Dynamic HTML.* Paris : Simo & Schuster Macmillan, 1998. ISBN 2-7440-0370-0.

Chambre de Commerce et d'Industrie de Paris, Direction de l'information Economique. *Le Thesaurus de DELPHES.* Sixième édition. Paris : Chambre de Commerce et d'Industrie de Paris, 1993. ISBN 2-85504-054-X.

COHEN Isaac. *CGI/Perl et JavaScript, Création de pages HTML interactives.* Paris : Edition Eyrolles, 1992. ISBN 2-212-08918-X.

FRANTZ Gérard. *Programmer en Visual Basic Script, Livre d'or.* SYBEX 1996. ISBN 2-7361-2269-0.

HOMER Alex, GILL Darren, JAKAB Stephen. *Interface entre Web et bases de données sous Windows NT. Internet Information Server Active Server Pages et objets ADO.* Traduit et adapté de l'anglais par Marc ISRAEL. Paris : Edition Eyrolles, 1998. ISBN 2-212-08964-3.

JEUGE-MAYNART Isabelle (Direction), **MARQUET Catherine** (Direction éditoriale), **FAUVEL Jean-Jacques** (Édition), **et al.** *Découvrir la France, (Parcs, Musées & Châteaux, Promenades, Patrimoine, Cartes & Plans).* Première édition. Paris : Hachette Tourisme, 1998. ISBN : 2-01-242871-1.

LECOMTE Cyrille, LEDUC Thomas. *Programmation Java Script.* Paris : Editions Eyrolles, 1997. FiSystem. Direction de : THÉVENEZ Thierry. ISBN 2-212-08937-6.

LOUVET Gérald. *Se former à Merise : la modélisation conceptuelle.* Collaboration de MAÎTRE François. Paris : Les Editions d'Organisation, 1990. Ingénierie des systèmes d'Information. Direction de : Hubert TARDIEU. ISBN 2-7081-1146-9

MOURIÈS Nathalie (Direction). *Guide des Villages de Charme en France.* Photos de couverture de SARRAMON Christian. 5ème édition. Paris : Editions Payot & Rivages, 1994. ISBN : 2-86930-743-8.

NAUGHTON Patrick. *The JAVA Handbook, The Authoritative Guide to the Java Revolution.* Berkeley, California : Osborne McGraw-Hill, 1996. ISBN : 0-07-882199-1.

- *Parties de monographies*

La BNF (Bibliothèque Nationale de France), Service de coordination bibliographique et le Ministère de l'enseignement Supérieur et de la Recherche, ABES (Agence Bibliographique de l'Enseignement Supérieur).
RAMEAU, Répertoire d'Autorité - Matière Encyclopédique et Alphabétique Unifié.
 Vol. 4. Montpellier : ABES 1997. p. 316-317.

La BNF (Bibliothèque Nationale de France), Service de coordination bibliographique et le Ministère de l'enseignement Supérieur et de la Recherche, ABES (Agence Bibliographique de l'Enseignement Supérieur).
RAMEAU, Répertoire d'Autorité - Matière Encyclopédique et Alphabétique Unifié.
 Vol. 6. Montpellier : ABES 1997. P. 280-282, 553-555.

- *Liste de discussion*

Liste de discussion de servlets. [On-Line]. U.S. : Sun Microsystems, Inc., August 1999. Available from Internet : <URL : listserv@java.sun.com >

- *Sites Internet:*

- *Annuaire de villes et villages de France*

<http://www.communes.com/>

<http://www.annu-ville.com/>

http://www.yahoo.fr/Exploration_geographique/Pays/France/Villes/

<http://perso.wanadoo.fr/entreprise/>

- *Programmation (logiciels):*

<http://www.java.sun.com/>

<http://www.gamelan.com>

<http://deis187.deis.unibo.it/users/radu/netmaps/>

▪ *Adresses électroniques :*

Bartholo Phil . *The Word Spreads : developers abandon CGI in favor of Servlets.* [On-line]. U.S. : Sun Microsystems, Inc., April 1999. [23.08.1999]. Available from Internet : <URL : <http://www.java.sun.com/features/1998/03/inetbiz.html> >

Java Servlet API. [On-line]. U.S. : Sun Microsystems, Inc., September 1999. [18.08.1999]. Available from Internet :
<URL : <http://www.java.sun.com/products/servlet/index.html> >

The official directory for Java. NetMaps (Software: Applet). [On-line]. USA : developer.com, directories: gemelan and Sun Microsystems, January 1999 [05.08.1999]. Available from Internet : <URL : <http://www-a.developer.com/directories/pages/dir.java.util.maps.html> >

Villages et Villes de France. Webring . [On-line]. [10.06.1999]. Available from Internet : <URL : <http://www.webring.org/cgi-bin/webring?ring=v2f:list> >