

## TABLES DES MATIERES

REMERCIEMENT

TABLES DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ABREVIATIONS

INTRODUCTION .....	1
PARTIE I : .....	2
GENERALITES .....	2
CHAPITRE I. GENERALITES .....	3
I- Quelques Définitions .....	3
1. Notion de SIG.....	3
2. Notion de cartographie .....	4
3. Notion de Base de données.....	4
4. Système de gestion de base de données .....	4
5. Web .....	5
6. Webmapping .....	5
II- Logiciel libre .....	6
1. L'Open Source .....	6
2. L'OGC .....	7
III- Solutions et technologies existantes .....	7
1. Coté client.....	7
2. Côté serveur.....	9
3. Services web Geospatiaux.....	10
4. Format et technologie de publication de données spatiales sur le web .....	11
IV- Critères d'évaluation pour le choix des logiciels utilisés.....	12
V- SOLUTIONS.....	12

1.	Les solutions retenues .....	12
VI-	QGIS .....	13
1.	Définition.....	13
2.	Fonctionnalité.....	13
3.	Barre de Menu .....	14
4.	Barre d'outils.....	15
5.	Légende de la carte.....	15
6.	Affichage de la carte .....	16
7.	Barre d'état.....	16
	CHAPITRE II. ETUDE PRELIMINAIRE .....	17
I-	Planning et organisation .....	17
II-	Planning prévisionnel.....	18
III-	Recueil des besoins du système .....	18
1.	Fonctionnalités du système .....	18
2.	Besoins opérationnels ou techniques .....	19
3.	Identification des messages .....	19
IV-	Analyse et étude conceptuelle .....	20
1.	Identification des cas d'utilisation .....	20
	PARTIE II : .....	22
	MISE EN PLACE DE LA BASE DE DONNEES .....	22
	CHAPITRE I. MISE EN PLACE DE LA BASE DE DONNEES.....	23
I-	PostgreSQL/PostGIS .....	23
1.	PostgreSQL .....	23
2.	PostGIS.....	23
II-	Mise en place de la base de données.....	23
1.	Configuration de PostgreSQL et de sa cartouche spatiale PostGis.....	23
2.	L'organisation d'une base de données .....	24
3.	Création de la structure de la base de données .....	24
4.	Langage.....	26

III- Méthodologie de conversion des données .....	28
PARTIE III : .....	30
LA SOLUTION WEBMAPPING .....	30
CHAPITRE I. LE WEBMAPPING .....	31
I- Notion sur le webmapping .....	31
1. Définition .....	31
2. Utilité .....	31
3. Contexte .....	31
4. Principes et fonctionnement du webmapping .....	32
5. Architecture d'une application web .....	32
6. Architecture d'une application webmapping .....	33
II- MISE EN ŒUVRE DE LA PLATE-FORME .....	34
III- ENVIRONNEMENT ET OUTILS UTILISES .....	34
1. Mapserver .....	34
2. Mapserver et Apache .....	34
3. PostgreSQL .....	36
4. PostGIS .....	37
IV- Langages utilisés .....	39
1. HTML .....	39
2. JavaScript .....	40
3. PHP .....	40
CHAPITRE II. REALISATIONS .....	42
I- Mise en place du serveur cartographique avec Mapserver .....	42
1. Création et configuration du mapfile .....	43
2. Développement de l'interface cartographique .....	47
CONCLUSION .....	49
GLOSSAIRE .....	50
Bibliographie .....	51
ANNEXES .....	53

Annexe 1 : Installation de PostgreSql/PostGIS .....	54
Annexe 2 : Installation de Mapserver et Pmapper .....	59
Annexe 3 : Description du mapfile .....	60
Annexe 4 : Installation de QGIS 2.8.....	76

## LISTE DES FIGURES

Figure 1 : Mode de représentation vecteur

Figure 2 : Mode de représentation raster

Figure 3 : L'interface de QGIS

Figure 4 : Organisation d'une base de données

Figure 5 : Interface de création d'une table par la méthode SQL

Figure 6 : Création d'une table par la méthode directe

Figure 7 : Ajout d'une table PostGIS

Figure 8 : Architecture d'une application web

Figure 9 : Architecture d'une application de webmapping

Figure 10 : Mise en évidence des applications Web et PostGIS/PostgreSQL

Figure 11 : Interface d'une page HTML

Figure 12 : Place du PHP sur le serveur WEB

Figure 13 : Fenêtre de démarrage de MapServer

Figure 14 : Fenêtre de connexion à Postgis

Figure 15 : Vue de la couche sous QGis

Figure 16 : Erreur Python

Figure 17 : Export de la couche vers MapServer

Figure 18 : Aperçus dans PMAPPER

## **LISTE DES TABLEAUX**

Tableau 1 : Logiciels SIG bureautique

Tableau 2 : Logiciels SIG serveur

Tableau 3 : Formats et technologies de diffusion de données géographiques sur le web

Tableau 4 : Planning prévisionnel détaillé

Tableau 5 : Récapitulatif des cas d'utilisation avec les acteurs impliqués

Tableau 6 : Paramètre de connexion PostGIS

## LISTE DES ABREVIATIONS

**BD** : Base de Données  
**CGI** : Common Gateway Interface  
**CSS** : Cascading Style Sheets  
**ESPA** : Ecole Supérieure Polytechnique d'Antananarivo  
**GML** : Geography Markup Language  
**GPL** : General Public License  
**GPS** : Global Positioning System  
**Http** : HyperText Transfer Protocol  
**HTML** : HyperText Markup Language  
**IGAT** : Information Géographique et Aménagement du Territoire  
**LMD** : Langage de manipulation de données  
**LCD** : Langage de contrôle de données  
**MCD** : Model Conceptuel de Données  
**MLD** : Model Logique de Données  
**MPD** : Model Physique de Données  
**MS4W** : MapServer for Windows  
**OGC** : Open Geospatial Consortium  
**OSGeo4W** : OSGeo for Windows  
**OSM** : Open Street Maps  
**PHP** : Personal Home Page  
**POO** : Programmation Orientée Objet  
**SGBD** : Système de Gestion de Base de Données  
**SHP** : Shape  
**SIG** : Système d'Information Géographique  
**SQL** : Structured Query Language  
**SFS** : Simple Feature For SQL  
**UML** : Unified Modeling Language  
**URL** : Uniform Ressource Locator  
**WMS** : Web Map Service  
**WFS** : Web Feature Service  
**WCS** : Web Coverage Service  
**WWW** : World Wide Web  
**XML** : eXtensible Markup Language

## INTRODUCTION

Actuellement la technologie progresse avec une vitesse incroyable, le SIG (*Système d'Information Géographique*) pourrait énormément contribuer dans différent domaine de la vie quotidienne même dans ses moindres détails tel : trouver une pharmacie en cas d'urgence. Véritable outil d'aide, le SIG met à disposition de l'information localisée et cartographiée au grand public, et les technologies dans ce domaine n'ont cessé de se développer. Associé à l'outil SIG, l'Internet est un support de communication et de diffusion permettant de se communiquer et de s'échanger des informations géographiques. Cette communication entre ordinateurs permet plusieurs possibilités et offre une masse d'informations plus importante dans plusieurs domaines chaque jour.

Ce liaison SIG et Internet appelé « webmapping » est un domaine en pleine évolution ces dernières années. Et cette évolution s'est vue étoffée avec l'avènement du web 2.0 et surtout avec l'apparition des portails cartographiques tels que Google Maps, Bings Maps, OpenStreetMap et bien d'autres.

D'où le thème de ce mémoire : « Mise en place d'une application webmapping de géolocalisation de Pharmacie à Antananarivo ».

L'intérêt d'une telle plate-forme réside en sa capacité de fournir l'information géographique couplée à sa description pour mieux orienter le citoyen dans la ville.

L'étude consistera à analyser les besoins des utilisateurs, à concevoir une solution en proposant des outils adaptés et à réaliser la plate-forme. Plus concrètement, nous aurons à construire une base de données qui va contenir les pharmacies à Antananarivo et créer une interface pour la consultation et tracer l'itinéraire le plus court.

Pour la réalisation de ce projet, nous allons voir dans la première partie les généralités, ensuite dans la deuxième partie la mise en place de base de données, la troisième partie donne la solution webmapping.

# PARTIE I : **GENERALITES**

## CHAPITRE I. GENERALITES

### I- Quelques Définitions

#### 1. Notion de SIG

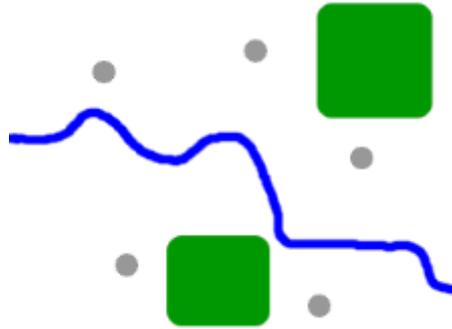
Les SIG sont utilisables pour aider la gestion des ; collectivités territoriales, secteur public, entreprise, écoles, administrations. La création de cartes et l'analyse géographique ne sont pas des procédés nouveaux, mais les SIG procurent une plus grande vitesse et proposent des outils sans cesse innovant dans l'analyse, la compréhension et la résolution des problèmes.

De nombreux autres domaines tels que la recherche et le développement de nouveaux marchés, l'étude d'impact d'une construction, l'organisation du territoire, la gestion de réseaux, le suivi en temps réel de véhicules, la protection civile... sont aussi directement concernés par la puissance des SIG pour créer des cartes, pour intégrer tout type d'information, pour mieux visualiser les différents scénarios, pour mieux présenter les idées et pour mieux appréhender l'étendue des solutions possibles.

Dans un SIG, deux types de données sont manipulés : les données attributaires (ou descriptives) et les données géographiques. Les données descriptives sont des données alphanumériques (quantitatives ou qualitatives) associées à un objet ou une localisation géographique, soit pour décrire un objet géographique, soit pour localiser des informations par exemple nom d'un hôtel, nombre d'étoile, nombre de chambre, nom du propriétaire. Les données géographiques sont les objets qu'on peut localiser à partir de leurs coordonnées géographiques. Nous avons deux modes de représentation de ces objets géographiques : le mode raster et le mode vecteur.

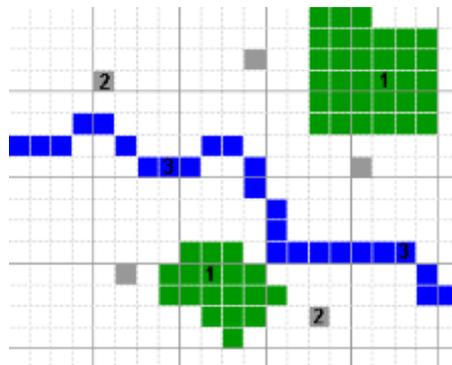
Le mode vecteur (figure1) est un mode de représentation géométrique sous forme de points, de lignes et de polygones. Dans ce mode, les informations sont regroupées sous la forme de coordonnées X, Y. Ainsi, les objets de type « point » (hôtels, restaurant, pharmacies...) sont représentés par les coordonnées X, Y ; les objets linéaires (routes, fleuves, ...) sont représentés par une succession de coordonnées X, Y et les objets polygonaux (les territoires...) par une succession de coordonnées X, Y délimitant une surface fermée. Le mode de représentation vecteur a pour avantage de donner une représentation très conforme à la réalité, de calculer avec précision la localisation et les dimensions des objets, d'individualiser les objets en leur attachant des attributs.

Le mode raster (figure2) correspond à une division régulière de l'espace sous forme de cellules ou mailles généralement carrées appelées pixels, qui définissent la précision minimale de la structure. Ce mode de représentation s'applique aux traitements d'images (satellites, photos aériennes). Il a pour avantage de faciliter le croisement des données et se prête bien à certains types de traitements numériques (convolutions, filtres...) et d'analyse.



**Figure 1 : Mode de représentation vecteur**

(Source : [http://www.notre-planete.info/terre/outils/sig\\_1.php](http://www.notre-planete.info/terre/outils/sig_1.php))



**Figure 2 : Mode de représentation raster**

(Source : [http://www.notre-planete.info/terre/outils/sig\\_1.php](http://www.notre-planete.info/terre/outils/sig_1.php))

## 2. Notion de cartographie

La cartographie est la discipline qui regroupe « l'ensemble des études et des opérations scientifiques, artistiques et techniques, intervenant à partir des résultats d'observations directes ou de l'exploitation d'une documentation, en vue de l'élaboration et de l'établissement de cartes, plans et autres modes d'expression, ainsi que dans leur utilisation » (Comité français de cartographie, 1990).

## 3. Notion de Base de données

Une Base de données (BD) est un ensemble structuré et organisé permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation (ajout, mise à jour, recherche de données).

## 4. Système de gestion de base de données

Un système de gestion de base de données (SGBD), parfois appelé uniquement gestionnaire de base de

données, est une solution qui permet à un ou plusieurs utilisateurs de créer et d'accéder à des données contenues dans une base de données. Le SGBD prend en charge les requêtes des utilisateurs ainsi que celles provenant d'autres programmes pour qu'ils n'aient pas à comprendre où la donnée se situe physiquement sur le support de stockage et, dans le cas d'un système multi-utilisateurs, qui d'autres peut également accéder aux données. En gérant les requêtes des utilisateurs, le SGBD garantit l'intégrité des données (s'assure qu'elles soient toujours disponibles et qu'elles soient organisées de façon cohérente) ainsi que la sécurité (s'assure que seulement ceux disposant des privilèges d'accès puissent accéder aux données). Le SGBD le plus courant est le système de gestion de base de données relationnelle (SGBDR). L'interface standard est le SQL (Structured Query Language). Le système de gestion de base de données orienté objets est une nouvelle forme de SGBD.

## 5. Web

Une Application Web (aussi appelée web app, de l'anglais) est une application manipulable grâce à un navigateur web (Mozilla Firefox, Google Chrome, Internet explorer, etc.). De la même manière que les sites web, une application web est généralement placée sur un serveur et se manipule en actionnant des widgets (composant logiciel en relation avec les interfaces graphiques) à l'aide d'un navigateur web, via un réseau informatique (Internet, intranet, etc.). Grâce à l'utilisation d'une SIG spécifié à la production d'une carte sur le web : « Webmapping », nous avons réussi à implémenter une carte dynamique et des outils convenables à la demande du Service Topographique dans l'application Web.

## 6. Webmapping

Le Webmapping (ou WebSIG) est la mise en ligne du système d'information géographique et plus largement de cartes permettant de diffuser celui-ci et celles-ci à travers le web (réseau internet). En d'autres termes, le webmapping ou la cartographie en ligne est l'usage des technologies de l'Internet pour le stockage et la diffusion de l'information géographique.

Une application de webmapping devrait pouvoir offrir les fonctionnalités ci-après :

- cartographier des données géographiques à la demande selon le choix des couches et de l'emprise géographique ;
- afficher des cartes dans un navigateur ;
- effectuer des mesures sur des cartes ;
- accéder à des bases de données métiers et sémantiques ;
- faire des recherches portant sur la sémantique ou la géométrie des données cartographiées ;
- saisir de l'information pour alimenter la base de données sur le serveur ;

- effectuer des traitements complexes comme le calcul des itinéraires ;
- imprimer des cartes en ligne ;

Aujourd'hui avec l'avènement du Web 2.0, le webmapping n'est plus au stade du simple affichage et de visualisation de cartes dans un navigateur. Mais, il se caractérise de nos jours par une forte interactivité et des contenus géolocalisés générés par les utilisateurs. Ainsi, chaque utilisateur peut créer des cartes personnalisées, les partager à d'autres utilisateurs.

Outre ces quelques notions, la réalisation de ce projet nécessite l'utilisation d'un logiciel.

## **II- Logiciel libre**

Les logiciels SIG utilisés sont bien l'élément le plus important dans les outils géomatiques, d'où on doit bien savoir choisir ces logiciels. Il faut cependant reconnaître qu'avec les années, un certain nombre de logiciel ont pu s'affirmer.

### **1. L'Open Source**

La désignation open source, ou « code source ouvert », s'applique aux logiciels dont la licence respecte des critères précisément établis par l'Open Source Initiative, c'est-à-dire les possibilités de libre redistribution, d'accès au code source et de création de travaux dérivés. Mis à la disposition du grand public, ce code source est généralement le résultat d'une collaboration entre programmeurs. L'open source a déjà investi tous les grands domaines du système d'information des administrations françaises : environnements serveurs, domaines applicatifs, outils d'ingénierie, solutions de réseaux et sécurité. Les solutions open source sont désormais au même rang que les solutions propriétaires dans le paysage des logiciels du secteur public. Les décideurs effectuent d'ailleurs de plus en plus leur choix à partir d'un jugement éclairé, en comparant systématiquement solutions propriétaires et solution libres.

## **2. L'OGC**

L'OGC (Open Geospatial Consortium) est l'organisme référent en matière de normalisation des informations géographiques. Association à but non lucratif, l'OGC élabore des normes pour le traitement de l'information géographique sur des plateformes informatiques ouvertes. Une de ses spécifications est de faciliter l'interopérabilité des systèmes afin de promouvoir le développement des SIG libres.

### **III- Solutions et technologies existantes**

#### **1. Coté client**

Coté client, plusieurs logiciels existent pour faire le SIG Web : ils sont dits logiciels « SIG web client lourd ». Ces logiciels offrent la gamme complète des fonctionnalités spécifiques aux SIG depuis la saisie et l'analyse des données jusqu'à la restitution cartographique. Ils possèdent des capacités de traitement évoluées et ont des interfaces graphiques sophistiquées. Ces outils permettent par exemple d'importer des données et de faire leur analyse, de modifier des données géométriques et attributaires, de se connecter à des sources de données distantes. Ci-dessous le tableau présentant quelques outils SIG bureautiques (Tableau 1).

Logiciels	Avantages	Inconvénients
Arcgis	<ul style="list-style-type: none"> <li>- Logiciel commercialisé par ESRI ;</li> <li>- Très complet dans l'analyse thématique ;</li> <li>- Connexion à Oracle Spatial, Microsoft Access, bases en WFS et WMS, etc.</li> </ul>	<ul style="list-style-type: none"> <li>- Interface assez conviviale ;</li> <li>- Installation difficile</li> </ul>
Mapinfo	<ul style="list-style-type: none"> <li>- Logiciel propriétaire ;</li> <li>- Permet d'importer et d'exporter de nombreux fichiers d'extension et de logiciels variés ;</li> <li>- Connexion à Oracle Spatial, Microsoft Access, bases en WFS et WMS, etc.</li> </ul>	<ul style="list-style-type: none"> <li>- Interface bureautique simple ;</li> <li>- Utilisation un peu complexe</li> </ul>
UDIG <sup>6(*)</sup> <i>User Friendly Desktop Internet Gis</i>	<ul style="list-style-type: none"> <li>- Outil libre ;</li> <li>- Édition de la géométrie des couches ;</li> <li>- Connexion WMS/WFS, PostGis, Oracle spatial ;</li> <li>- Très fort sur les connexions distantes.</li> </ul>	<ul style="list-style-type: none"> <li>- Nécessite une plate-forme Java JRE/JDK ;</li> </ul>
QGis <sup>7(*)</sup>	<ul style="list-style-type: none"> <li>- Logiciel libre ;</li> <li>- Visualisation, manipulation de données vectorielles et raster ;</li> <li>- Connexion WMS/WFS, PostGIS ;</li> <li>- Installation facile. Interface en français et conviviale. Simple d'utilisation.</li> </ul>	<ul style="list-style-type: none"> <li>- Plantage sur certains traitements</li> </ul>

**Tableau 1 :** Logiciels SIG bureautique

Aujourd'hui toutes les solutions SIG « clients bureautiques » ont la capacité d'offrir les services web OGC. Aussi, les développeurs de solutions de webmapping disposent d'APIs clientes qui permettent de se connecter aux services web OGC. C'est le cas de l'API propriétaire Google Maps<sup>8(\*)</sup>, mais aussi de l'API libre OpenLayers qui permet d'invoquer tous les services OGC.

## 2. Côté serveur

De nombreuses solutions existent côté serveur. Ces outils permettent la mise en ligne et la diffusion de l'information géographique à travers le réseau : intranet ou internet. Les utilisateurs se connectent à ses outils par l'intermédiaire des outils SIG bureautique, de navigateurs web, d'applications spécialisées, de matériels nomades, etc. Ci-dessous le tableau présentant quelques serveurs cartographiques (Tableau 2).

Logiciels	Descriptions
MapServer <sup>9(*)</sup>	<ul style="list-style-type: none"> <li>- Solution open source</li> <li>- Visualisation, manipulation de données vectorielles et raster ;</li> <li>- Fonctionnalités web ;</li> <li>- Connexion à postgis et aux services WMS, WFS, WCS ;</li> <li>- Installation facile.</li> </ul>
GeoServer <sup>10(*)</sup>	<ul style="list-style-type: none"> <li>- Logiciel OpenSource;</li> <li>- Visualisation, manipulation de données vectorielles et raster</li> <li>- Fonctionnalités web ;</li> <li>- Supporte plusieurs bases de données (PostgreSql/Postgis, oracle/oracle spatiale, ...) ;</li> <li>- Connexion aux services WMS, WFS et WCS ;</li> <li>- Nombreux formats de diffusion : JPEG, PNG, KML, GML, PDF, GeoJSON, SHP ;</li> <li>- Interface de gestion des données.</li> </ul>
ArcGis server	<ul style="list-style-type: none"> <li>- Serveur SIG commercialisé par ESRI ;</li> <li>- Solution complète ;</li> <li>- Dispose de sa propre terminologie de services géographiques ;</li> <li>- Indissociable d'ArcGis bureautique.</li> </ul>

**Tableau 2** : Logiciels SIG serveur

### 3. Services web Geospatiaux

Les services web géospatiaux offrent des solutions d'accès à des sources de données réparties sur un réseau. Ces services permettent aux systèmes d'accéder à des données grâce à des méthodes, des règles de communication et une série de paramètres qu'ils définissent.

Afin de garantir l'interopérabilité des données et des traitements entre les différentes solutions SIG, l'OGC<sup>11</sup>(<sup>©</sup>) a élaboré plusieurs protocoles d'échanges de données géographiques compatibles avec les protocoles standards de services. Et ces standards OGC sont : WMS, WFS, WCS, CSW, etc.

#### 3.1. WMS (Web Map Service)

Le WMS est une spécification internationale de diffusion et d'utilisation de cartes dynamiques sur le Web. WMS permet de produire des cartes de données géoréférencées à partir de différents serveurs de données. Il définit un ensemble de serveurs cartographiques mis en réseau pour construire des cartes interactives. Les données sont délivrées sous différents formats : tiff, gif, jpeg, bmp, png, svg.

Un serveur WMS satisfait aux requêtes GetCapabilities, GetMap et GetFeature. Une requête GetCapabilities retourne les métadonnées qui décrivent le contenu du service et les paramètres acceptés c'est-à-dire les capacités du serveur. La requête GetMap retourne une image de la carte demandée dont les paramètres géospatiaux et dimensionnels ont été définis. Le GetFeature retourne des informations sur un objet représenté sur la carte.

#### 3.2. WFS (Web Feature Service)

Le WFS permet, au moyen d'une URL formatée, d'interroger des serveurs cartographiques afin de manipuler des objets géographiques (lignes, points, polygones...), contrairement au WMS qui permet la production de cartes géoréférencées à partir de serveurs géographiques. WFS propose des interfaces pour la description des manipulations de données sur des objets géographiques. Les opérations de manipulation de données permettent de :

- créer des nouveaux objets ;
- effacer des objets ;
- mettre à jour des objets ;
- prendre ou rechercher des objets sur la base de contraintes spatiales.

Un serveur WFS répond aux requêtes GetCapabilities, DescribeFeatureType, GetFeature, LockFeature, Transaction. La requête DescribeFeatureType permet de retourner la structure de chaque

entité susceptible d'être fournie par le serveur. La requête Transaction permet de modifier un objet (création, mise à jour, suppression). Le LockFeature permet de bloquer des objets lors d'une transaction.

### 3.3. WCS (Web Coverage Service)

WCS est un standard OGC qui permet la mise en ligne des données géographiques de type raster. Un serveur WCS doit répondre aux requêtes GetCapabilities, DescribeCoverage et GetCoverage. La requête DescribeCoverage renvoie une description complète pour chaque couverture et GetCoverage renvoie une couverture dans différents formats.

## 4. Format et technologie de publication de données spatiales sur le web

Les données géographiques publiées utilisent divers formats. Ci-dessous quelques formats de diffusion de ces données (Tableau 3).

Format/technologie	Description
GML ( <i>Geographic Markup Language</i> )	<ul style="list-style-type: none"> <li>- spécifié par l'OGC donc interopérable avec toutes les spécifications OGC ;</li> <li>- basé sur du XML pour la spécification de l'information géographique de type vecteur.</li> </ul>
KML ( <i>Keyhole Markup Language</i> )	<ul style="list-style-type: none"> <li>- format pour spécification de données géographiques publiables sur Google (format standard de Google Earth) ;</li> <li>- dérivé du XML ;</li> <li>- permet d'échanger des données géographiques en deux et trois dimensions.</li> </ul>
GeoRSS ( <i>Geographically Encoded Objects for RSS</i> )	<ul style="list-style-type: none"> <li>- permet d'ajouter l'information relative à la localisation dans les fichiers de type RSS <sup>12(*)</sup>(Really Simple Syndication) ;</li> <li>- deux types d'implémentations : GeoRSS simple et GeoRSS GML.</li> </ul>
GeoJSON ( <i>Geographic JavaScript Object Notation</i> )	<ul style="list-style-type: none"> <li>- utilise la notation des objets Javascript ;</li> <li>- supporte les géométries : Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon et GeometryCollection.</li> </ul>
SVG ( <i>Scalable Vector Graphic</i> )	<ul style="list-style-type: none"> <li>- Langage de description graphique 2D en XML ;</li> </ul>

**Tableau 3 :** Formats et technologies de diffusion de données géographiques sur le web

La cartographie en ligne est en plein essor. Et les technologies dans ce domaine ne font qu'évoluer et aussi naître. Il reste que le monde du libre a occupé une place importante et propose de nombreux outils pour la diffusion et le partage de l'information géographique.

#### **IV- Critères d'évaluation pour le choix des logiciels utilisés**

On a choisi ces critères de telle sorte qu'ils séquent l'analyse de chaque logiciel libre en quatre étapes permettant au final d'être le plus objectif possible.

L'étude porte sur les caractéristiques générales, l'installation, l'interface et l'utilité du logiciel libre testé. Chaque étape se succédant, les solutions ne répondant pas aux premiers critères ne passeront pas aux étapes suivantes.

Les caractéristiques générales présentent, la version et le statut du logiciel, date de la dernière version téléchargeable, le type de logiciel fonctionnalités promises par le site, les formats supportés (vectoriels, matriciels) en entrée et en sortie, la licence d'utilisation précisant des conditions particulières

L'installation est décrite par, l'abondance et la langue de la documentation (éliminatoire s'il n'y a ni de la documentation en anglais, ni en français), de quel côté s'effectue l'installation, si l'application nécessite un serveur particulier (critère éliminatoire), le niveau requis des personnes tenues de la réaliser (il a été fixé selon les difficultés rencontrées pour faire fonctionner l'application), les extensions requises (on considère que c'est un critère négatif pour une application de nécessiter beaucoup d'extensions)

#### **V- SOLUTIONS**

##### **1. Les solutions retenues**

Au final, Quantum Gis a été retenu pour la large gamme de formats qu'il accepte en entrée et qu'il peut exporter. Sa documentation est abondante, il est facile d'installation et ne prend pas trop de place sur le disque dur. Son interface, en français, est conviviale et propose de nombreuses fonctionnalités parmi lesquelles la consultation, la mise à jour et l'édition de données liées à une base de données distante. En plus, il permet d'éditer des mapfiles (forme de fichier sur mapserver).

## **VI- QGIS**

### **1. Définition**

Quantum GIS est un SIG libre qui a débuté en mai 2002 et s'est établi en tant que projet en juin 2002 sur Source Forge. QGIS est utilisable sur la majorité des Unix, Mac OS X et Windows. Il utilise la bibliothèque logicielle Qt et le langage C++, ce qui se traduit par une interface graphique simple et réactive. Aussi, il se veut simple à utiliser, fournissant des fonctionnalités courantes. Le but initial était de fournir un visionneur de données SIG. QGIS a, depuis, atteint un stade dans son évolution où beaucoup y recourent pour leurs besoins quotidiens. Il supporte un grand nombre de formats raster et vecteur, avec le support de nouveaux formats facilité par l'architecture des modules d'extension. De plus, il est distribué sous la licence GPL (General Public License). Ceci signifie qu'on pouvait étudier et modifier le code source, tout en ayant la garantie d'avoir accès à un programme SIG non onéreux et librement modifiable. Nous pouvons aussi créer, éditer, gérer et exporter des données vectorielles dans plusieurs formats.

### **2. Fonctionnalité**

QGIS permet ce qui suit :

- numérisation pour les formats gérés par OGR et les couches vectorielles de GRASS
- création et édition des fichiers de forme (shapefiles), des couches vectorielles de GRASS et des tables géométriques Spatialite.
- géoréférencement des images avec l'extension de géoréférencement
- importation, exportation du format GPX pour les données GPS, avec la conversion des autres formats GPS vers le GPX ou l'envoi, la réception directement vers une unité GPS
- visualisation et édition des données OpenStreetMap
- création de couches PostGIS à partir de fichiers shapefiles grâce au module d'extension SPIT
- prise en charge améliorée des tables PostGIS
- gestion des attributs des couches vectorielles grâce à l'extension de gestion des
- tables ou celle de tables attributaires
- enregistrer des captures d'écran en tant qu'images géoréférencées

### **Installation de QGIS (voir Annexe2)**

Quand QGIS démarre, l'interface se présente sous la forme affichée ci-dessous

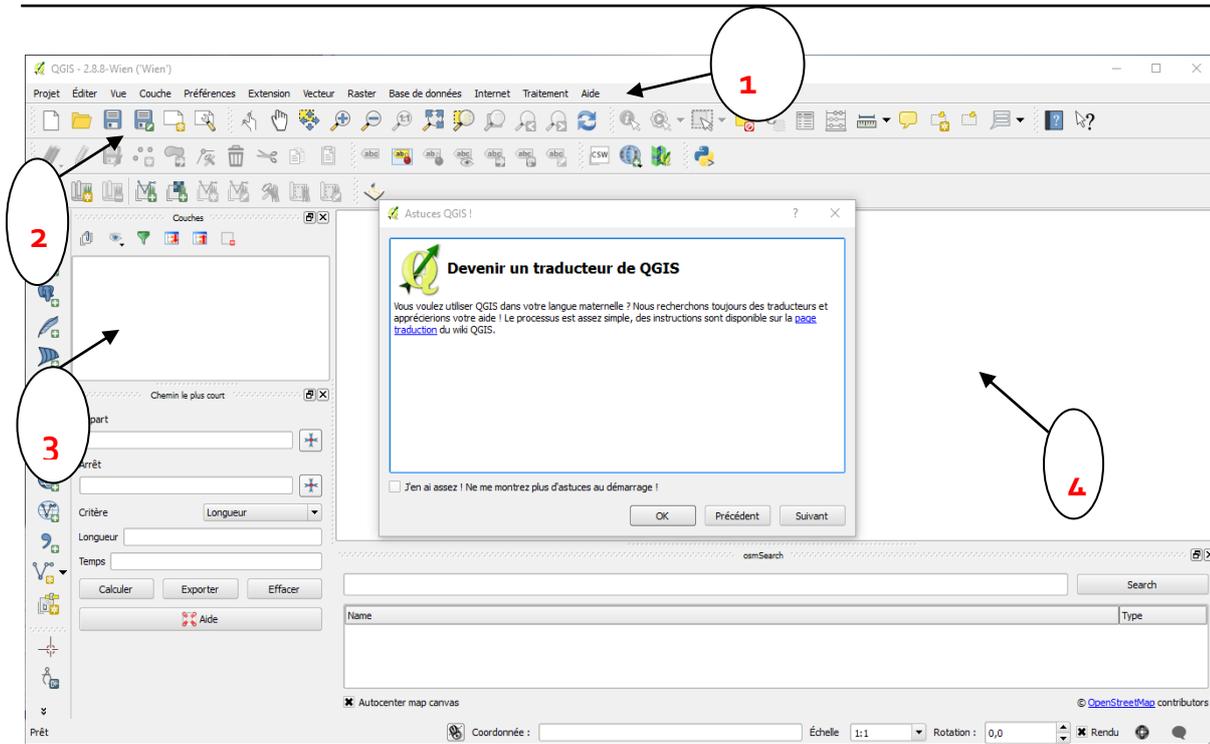


Figure 3 : L'interface de QGIS

L'interface de QGIS est divisée en cinq zones distinctes :

1. Barre de Menu
2. Barre d'Outils
3. Légende de la carte
4. Affichage de la carte
5. Barre d'état

### 3. Barre de Menu

La barre de menu donne accès aux différentes fonctionnalités de QGIS par le biais de menus hiérarchiques. Les entrées du menu de niveau supérieur et un résumé de certaines options sont listés ci-dessous, avec les icônes des outils correspondants dans la barre d'outils et leurs raccourcis clavier. Les raccourcis clavier présentés ici sont ceux définis par défaut mais ils peuvent également être configurés manuellement via le menu Préférences → Configurer les raccourcis....

Bien que les options de menu aient des outils qui leur correspondent et vice-versa, les menus ne sont pas organisés comme les barres d'outils. La barre contenant l'outil est affichée sous chaque option de menu en tant que case à cocher. Certaines entrées n'apparaissent que lorsque les extensions

correspondantes sont activées. Pour plus d'informations sur les outils et les barres d'outils, veuillez lire la section Barre d'outils.

### 3.1. Editer

Barre de Menu	Raccourci	Référence	Barre d'outils
Nouveau	Ctrl+N	voir <a href="#">Les projets</a>	Projet
Ouvrir	Ctrl+O	voir <a href="#">Les projets</a>	Projet
Nouveau depuis un modèle →			
Ouvrir un projet récent →		voir <a href="#">Les projets</a>	
Enregistrer	Ctrl+S	voir <a href="#">Les projets</a>	Projet
Enregistrer sous...	Ctrl+Shift+S	voir <a href="#">Les projets</a>	Projet
Enregistrer comme image...		voir <a href="#">Les projets</a>	Projet
Export DXF		voir <a href="#">Sortie graphique</a>	
Nouveau composeur d'impression	Ctrl+P	voir <a href="#">Sortie graphique</a>	Projet
Gestionnaire de composition ...		voir <a href="#">Composeur d'Impression</a>	
Composeurs d'impression →		voir <a href="#">Composeur d'Impression</a>	Projet
Fermer QGIS	Ctrl+Q	voir <a href="#">Composeur d'Impression</a>	

## 4. Barre d'outils

La barre d'outils fournit un accès à la majorité des fonctions des menus en plus d'outils additionnels destinés à interagir avec la carte. Chaque outil dispose d'une bulle d'aide qui s'affiche lorsque on place le curseur au-dessus. Celle-ci affiche une courte description du rôle de l'outil.

Chaque barre de menu peut être déplacée selon nos besoins. On peut les désactiver à partir du menu contextuel qui s'affiche d'un clic droit de la souris sur la barre d'outils

## 5. Légende de la carte

La partie de légende de la carte liste l'ensemble des couches du projet. La case à cocher dans chaque entrée de légende peut être utilisée pour afficher ou masquer la couche. La barre d'outils Légende dans la légende de la carte vous permet d'Ajouter un groupe, de Gérer la visibilité d'une couche pour toutes les couches disponibles ou de gérer une combinaison pré-établie de couches, de Filtrer la légende selon le contenu de la carte, de Tout étendre ou de Tout fermer et de Supprimer une couche ou un groupe de

couches.

## 6. Affichage de la carte

C'est la partie centrale de QGIS puisque les cartes y sont affichées ! Le contenu qui s'affiche dépend des couches de types raster et vecteur que vous avez choisies de charger (lire les sections suivantes pour plus d'informations sur comment charger une couche). L'emprise de la carte peut être modifiée en portant le focus sur une autre région, ou en zoomant en avant ou en arrière. Plusieurs opérations peuvent être effectuées sur la carte comme il est expliqué dans les descriptions des barres d'outils. La carte et la légende sont étroitement liées — la carte reflète les changements que vous opérez dans la légende.

On peut utiliser la molette de la souris pour changer le niveau de zoom de la carte. On place le curseur dans la zone d'affichage de la carte et faites rouler la molette vers l'avant pour augmenter l'échelle, vers l'arrière pour la réduire. La vue sera recentrée sur la position du curseur de la souris. On peut modifier le comportement de la molette de la souris en utilisant l'onglet Outils cartographiques dans le menu Préférences → Options.

On peut aussi utiliser les flèches du clavier pour vous déplacer sur la carte. Placez le curseur sur la carte et appuyez sur la flèche droite pour décaler la vue vers l'Est, la flèche gauche pour la décaler vers l'Ouest, la flèche supérieure vers le Nord et la flèche inférieure vers le Sud. On peut aussi déplacer la carte en gardant la touche espace appuyée et en bougeant la souris ou encore simplement en gardant la molette de la souris appuyée.

## 7. Barre d'état

La barre d'état montre votre position dans le système de coordonnées de la carte (coordonnées exprimées en mètres ou degrés décimaux par exemple) lorsque vous déplacez votre curseur. À gauche de l'affichage des coordonnées se trouve un petit bouton qui bascule l'affichage entre celui des coordonnées de la position ou celui de l'étendue de la zone que vous visualisez.

Quand on démarre QGIS, le degré décimal est l'unité par défaut. QGIS exprime les coordonnées de couches dans cette unité. Pour avoir les valeurs correctes d'échelle, on peut soit changer manuellement ce paramètre en mètres depuis l'onglet Général du menu Préférences → Propriétés du projet..., soit sélectionner un système de projection de référence en cliquant sur l'icône « SCR actuel » en bas à droite de la barre d'état.

## CHAPITRE II. ETUDE PRELIMINAIRE

L'étude préliminaire sont d'une aide très utile lors d'une réalisation d'un projet. Ils nous guident sur les étapes à franchir afin d'arriver aux résultats attendus du projet. Dans notre cas, la planification et l'organisation, la définition des besoins, la documentation, l'analyse et étude conceptuelle, collecte de données et l'inventaire de ces données constituent ces travaux.

### I- **Planning et organisation**

L'objectif est de bien répartir les tâches suivant un délai prévu pour la réalisation du projet. Ainsi, la durée de réalisation de ce projet a été prévue dans six mois.

Ainsi, les pourcentages sur les travaux ont été subdivisés en quatre grandes étapes, à savoir :

La documentation qui occupe 20% du temps,

L'Analyse fonctionnelle/Etude technique 10%

Le développement qui on nécessite 50%

Le reste a été consacré à la rédaction et à la préparation de la soutenance, autrement dit, le 20% du temps.

## II- Planning prévisionnel

Étapes	Délai	Activités
Cadrage du projet	Janvier	- Recueil des besoins - Elaboration du cahier des charges
Analyse fonctionnelle/Etude technique	Février	- Détermination des cas d'utilisations fonctionnels ; - Détermination des cas d'utilisations techniques ; - Modélisation du système (diagramme des cas d'utilisation, de séquences et de classes) ;
Conception de la solution	Février- Mars	- Elaboration du modèle de déploiement ; - Détermination et conception des interfaces ;
Réalisations/Implémentation de la solution	Mars Avril	- Codage des composants de l'application ; - Test
Rédaction du mémoire et soutenance	Avril Mai	- Assemblage des notes pour rédiger le mémoire - Préparation de la présentation

**Tableau 4 : Planning prévisionnel détaillé**

## III- Recueil des besoins du système

S'inspirant des solutions webmapping de géolocalisation existante et après une étude auprès des utilisateurs que sont les responsables ou gestionnaires des points d'intérêt, nous avons pu recueillir les besoins ci-dessous.

### 1. Fonctionnalités du système

Notre système doit pouvoir offrir les différentes fonctions ou services que sont :

- la navigation sur une interface cartographique simple ;
- la géolocalisation des points d'intérêt directement sur l'interface cartographique ;

- le calcul des itinéraires ;
- la recherche des lieux d'intérêt via un moteur de recherche ;
- la consultation des informations sur les points d'intérêt ;

## **2. Besoins opérationnels ou techniques**

La mise en place de la plate-forme doit tenir compte d'un certain nombre de besoins techniques dont :

- la sécurité d'accès au système : il faut garantir que les utilisateurs qui accèdent au système ont effectivement les droits requis. Un administrateur système est chargé de définir les profils des utilisateurs ;
- l'accessibilité : l'application doit être accessible via une interface web à partir d'un poste connecté à Internet ;
- la rapidité d'accès : le système doit pouvoir répondre aux demandes des utilisateurs en temps réel ;
- les Fonds de carte : utilisation de fonds de carte Google maps, OpenStreet maps et de fonds de carte constitué par nous-même.

## **3. Identification des messages**

Un message représente la spécification d'une communication unidirectionnelle entre objets et qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur. Un message est normalement associé à deux occurrences d'évènement : un évènement d'envoi et un évènement de réception. Les messages répertoriés entre le système et ses acteurs sont :

### **• Messages émis par le système**

- les affichages de la carte ;
- les localisations des points d'intérêt sur la carte ;
- les tracés des itinéraires ;
- les confirmations lors des validations des données.

### **• Messages reçus par le système**

- la création, modification et suppression de points d'intérêt ;
- la recherche de points d'intérêt ;
- la visualisation de points d'intérêt ;
- les informations relatives aux points d'intérêt.

## **IV- Analyse et étude conceptuelle**

Cette partie nous permettra de faire l'analyse de notre système. Cette analyse consiste dans un premier temps à rechercher et à décrire les cas d'utilisation de notre système à travers le diagramme de cas d'utilisation et les diagrammes de séquences. Cette analyse nous conduira aussi à la production de diagramme de classes et à la conception du modèle de stockage des données.

### **1. Identification des cas d'utilisation**

Un cas d'utilisation (use case) représente un ensemble de séquences d'actions réalisées par le système et qui produit un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l'acteur concerné. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera. L'ensemble des cas d'utilisation doit décrire exhaustivement les exigences fonctionnelles du système.

#### **a- Cas d'utilisation répertoriés**

Les cas d'utilisation répertoriés en fonction des différents acteurs sont :

- naviguer ou Visualiser la carte ;
- localiser un objet sur la carte ;
- consulter les informations liées à un point d'intérêt ;
- rechercher un point d'intérêt ;
- calculer des itinéraires ;
- inscrire les points d'intérêt ;
- mettre à jour les données ;
- valider les données d'un point d'intérêt ;
- administrer la plate-forme.

Ci-dessous le tableau récapitulatif des cas d'utilisation.

<b>Cas d'utilisation</b>	<b>Acteurs impliqués</b>
Visualiser carte	Internaute, serveurs de fond de carte
Localiser point d'intérêts	Internaute, serveur de fond de carte
Consulter les informations liées à un point d'intérêt	Internaute, serveur de fond de carte
Rechercher un point d'intérêt	Internaute, serveur de fond de carte
Calculer des itinéraires	Internaute, serveur de fond de carte
S'inscrire	Internaute
Inscrire un point d'intérêt	Gestionnaire de point d'intérêt
Mettre à jour un point d'intérêt	Gestionnaire de point d'intérêt
Valider un point d'intérêt	Administrateur
Administrer la plate-forme	Administrateur

**Tableau 5** : Récapitulatif des cas d'utilisation avec les acteurs impliqués

**PARTIE II :**  
**MISE EN PLACE DE LA BASE  
DE DONNEES**

## CHAPITRE I. MISE EN PLACE DE LA BASE DE DONNEES

Dans cette deuxième partie on va parler de l'intégration de la base de données et la connexion vers QGIS. Il faut noter que je n'ai pas créer la base de données, mais la base de données a été déjà créée par mon binôme sur la réalisation de ce projet.

### I- PostgreSQL/PostGIS

#### 1. PostgreSQL

PostgreSQL est un SGDB (Système de Gestion de Base de Données). C'est un logiciel capable d'enregistrer et de conserver des informations numériques et de les restituer à la demande d'un utilisateur. Les concurrents de PostgreSQL les plus courants sont Oracle, Sybase, Informai, Internasse, etc....., mais l'avantage de PostgreSQL par rapport aux autres SGDB est sa gratuité, alors que des systèmes tel qu'Oracle ne sont abordable que par de très grosse société. PostgreSQL est toutefois un peu plus rudimentaire que ses compagnons qui possèdent énormément d'outils d'aide à l'utilisation ou développement. Il possède leurs caractéristiques principales et essentielles.

#### 2. PostGIS

PostGIS est un module pour la base de données PostgreSQL permettant de rajouter le support des données spatiales directement dans la base de données, en termes de stockage, de manipulation et de requête. Il implémente la norme SFS (Simple Feature for SQL) de l'OGC tout en proposant également de nombreuses fonctionnalités additionnelles. De plus, PostGIS ajoute au système ses types de données et ses fonctions, mais aussi deux tables utilitaires : geometry\_columns et spatial\_ref\_sys. La première sert à indiquer au logiciel quels sont les champs contenant des types géographiques dans chacune des tables. La seconde contient les paramètres des systèmes de projection supportés et sert en interne au logiciel. A remarquer que le système de projection Laborde pour Madagascar y figure aussi.

### Installation de PostgreSQL/PostGIS (voir Annexe1)

### II- Mise en place de la base de données

#### 1. Configuration de PostgreSQL et de sa cartouche spatiale PostGis

Une fois le fichier d'installation version Windows de PostgreSQL téléchargé, l'installation est aisée en suivant la procédure d'installation pas à pas. Après installation de PostgreSQL, sa couche spatiale PostGis est installée à partir d'un exécutable téléchargé. Cette installation de PostGis peut être aussi faite avec l'application « Stack Builder » proposée par PostgreSQL. Les bases de données « postgis »

et « template\_postgis\_20 » sont ainsi créées dans PostgreSQL ce qui lui permet de gérer les données géographiques.

## 2. L'organisation d'une base de données

Le serveur est organisé en plusieurs bases de données

Chaque base de données l'est en schémas (par défaut chacune a un schéma « public »)

Les schémas contiennent des tables

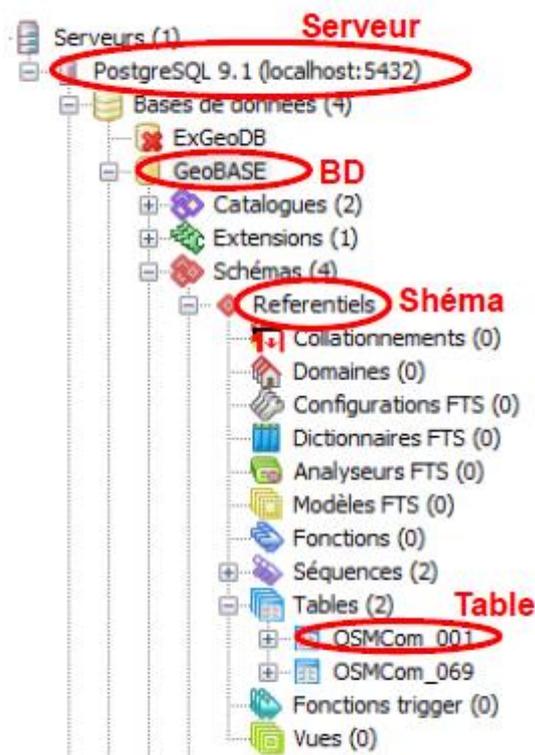


Figure 4 : Organisation d'une base de données

## 3. Création de la structure de la base de données

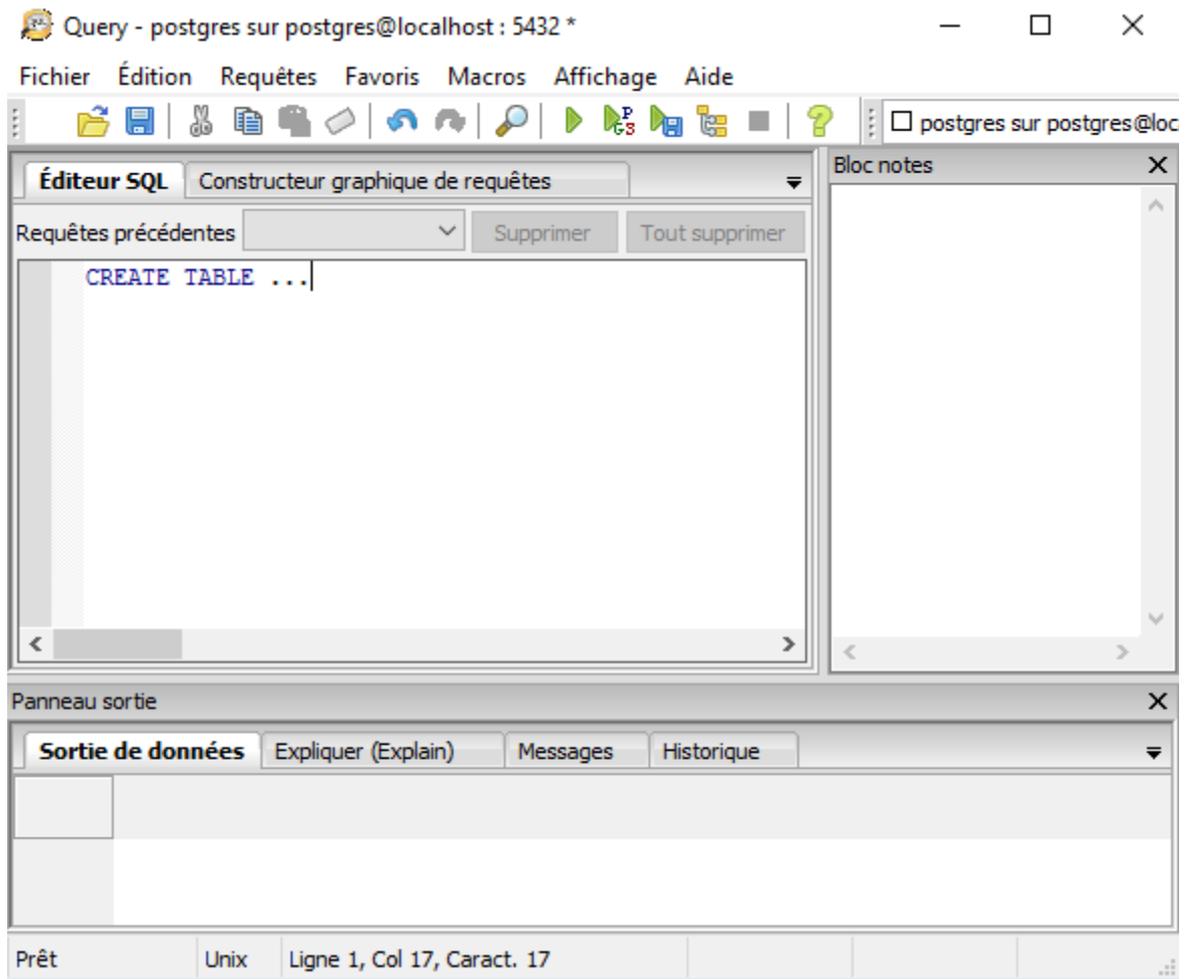
Après modélisation des données qui a abouti au diagramme de classes d'où nous avons tiré le modèle relationnel pour le stockage des données, vient ensuite l'implémentation de la base de données. L'implémentation de la base de données a consisté à créer notre base de données, puis les différentes tables du schéma relationnel et de définir les types de données des attributs et les contraintes. Nous avons réalisé ce travail avec l'outil d'administration PgAdmin III de PostgreSQL 9.4.

L'insertion de données dans PgAdmin peut se réaliser par deux méthodes :

- Ligne de commande SQL
- Interface graphique

### a- Ligne de commande SQL

La méthode SQL s'effectue en cliquant sur l'icône « SQL » après avoir sélectionné la table qui figure dans schéma et public.



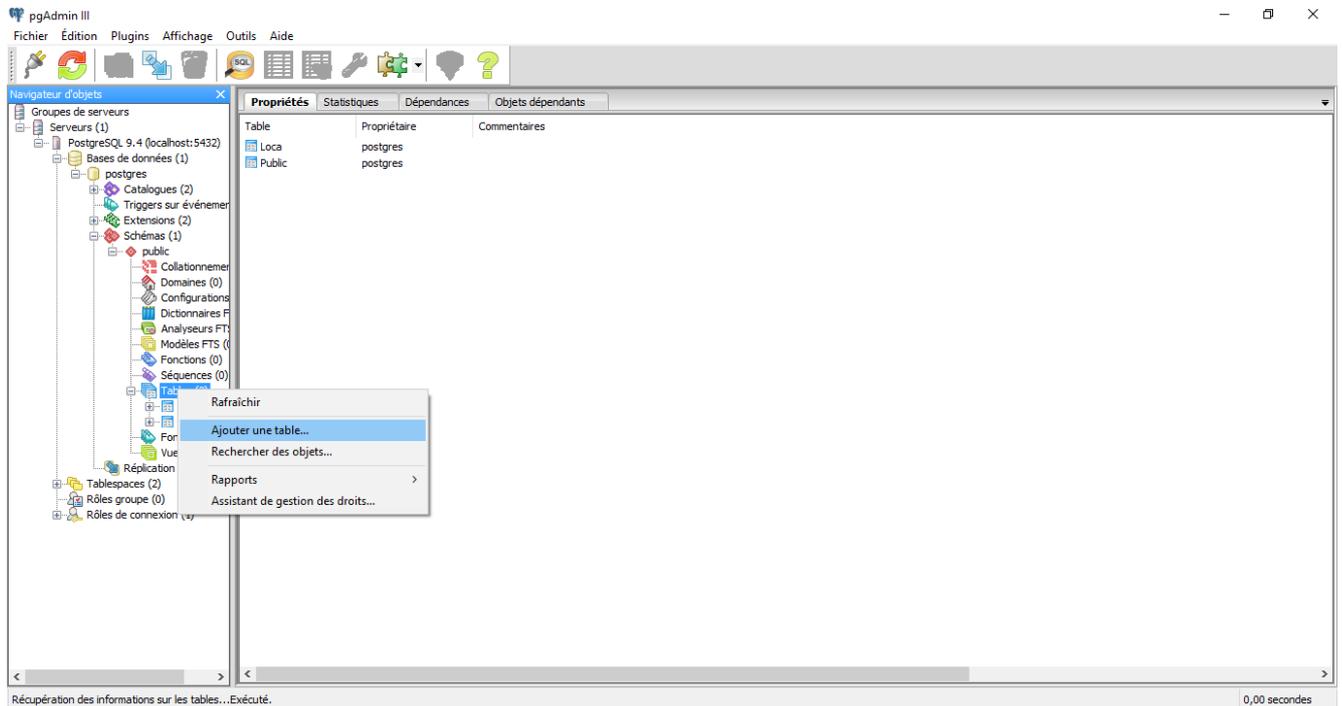
**Figure 5 :** Interface de création d'une table par la méthode SQL

CREATE TABLE pour créer la table. Par exemple, pour créer une table commune, on a :

```
CREATE TABLE commune  
  
idcommune serial,  
  
...
```

## b- Interface graphique

La méthode directe consiste à faire une cliquer droite sur la table, puis, sur ajouter une table



**Figure 6** : Création d'une table par la méthode directe

## 4. Langage

Le langage de requête utilisé dans ce SGBD est le SQL. SQL (Structured Query Language, ou Langage de requêtes structuré) est un langage de définition de données (LDD, ou en anglais DDL Data Definition Language), un langage de manipulation de données (LMD, ou DML : Data Manipulation Language), et un langage de contrôle de données (LCD, ou DCL : Data Control Language), pour les bases de données relationnelles.

### 4.1 Syntaxe de la commande SELECT

La commande SELECT est basée sur l'algèbre relationnelle, en effectuant des opérations de sélection de données sur plusieurs tables relationnelles par projection. Sa syntaxe est la suivante :

```
SELECT [ALL] | [DISTINCT] <liste des noms de colonnes>
```

```
FROM <Liste des tables>
```

```
[WHERE <condition logique>]
```

Il existe d'autres options pour la commande SELECT comme :

GROUP BY

HAVING

ORDER BY

#### 4.2 Insertion de données

L'insertion de nouvelles données dans une table se fait grâce à l'ordre INSERT, qui permet d'insérer de nouvelles lignes dans la table.

L'ordre INSERT attend la clause INTO, suivie du nom de la table, ainsi que du nom de chacune des colonnes entre parenthèses (les colonnes oubliées prendront la valeur NULL par défaut). Les valeurs à insérer peuvent être précisées de deux façons :

- avec la clause VALUES: une seule ligne est insérée, elle contient comme valeurs, l'ensemble des valeurs passées en paramètre dans la parenthèse qui suit la clause VALUES
- Les données sont affectées aux colonnes dans l'ordre dans lequel les colonnes ont été déclarées dans la clause INTO ; INSERT INTO Nom de la table (colonne1, colonne2, colonne3, ...) ; VALUES (Valeur1, Valeur2, Valeur3, ...).

#### 4.3 Modification de données

La modification de données aussi appelée mise à jour consiste à modifier des tuples (des lignes) dans une table grâce à l'ordre UPDATE. La modification à effectuer est précisée après la clause SET. Il s'agit d'une affectation d'une valeur à une colonne grâce à l'opérateur = suivi d'une expression algébrique, d'une constante ou du résultat provenant d'une clause SELECT. La clause WHERE permet de préciser les lignes sur lesquels la mise à jour aura lieu. La structure est représentée comme suit :

UPDATE Nom\_de\_la\_table

SET Colonne = Valeur\_Ou\_Expression

WHERE qualification

#### 4.4 Suppression de données

La suppression de données dans une table se fait grâce à l'ordre DELETE. Celui-ci est suivi de la clause FROM, précisant la table sur laquelle la suppression s'effectue, puis d'une clause WHERE qui décrit la qualification, c'est-à-dire l'ensemble des lignes qui seront supprimées.

DELETE FROM Nom\_de\_la\_table

WHERE qualification

L'ordre DELETE est à utiliser avec précaution car l'opération de suppression est irréversible. Il faudra donc s'assurer dans un premier temps que les lignes sélectionnées sont bien les lignes que l'on désire supprimer.

### III- Méthodologie de conversion des données

Après avoir intégré les données dans Postgresql. Ainsi, une méthodologie de conversion et de transformation de ces données est mise en place, utilisant le logiciel Quantum Gis.

#### Processus de Mise à jour

Pour réaliser la connexion entre le client lourd et le serveur de base de données, nous devons créer une connexion enregistrée en cliquant sur le bouton « Ajouter une couche PostGIS » de la barre d'outils de l'interface principale.

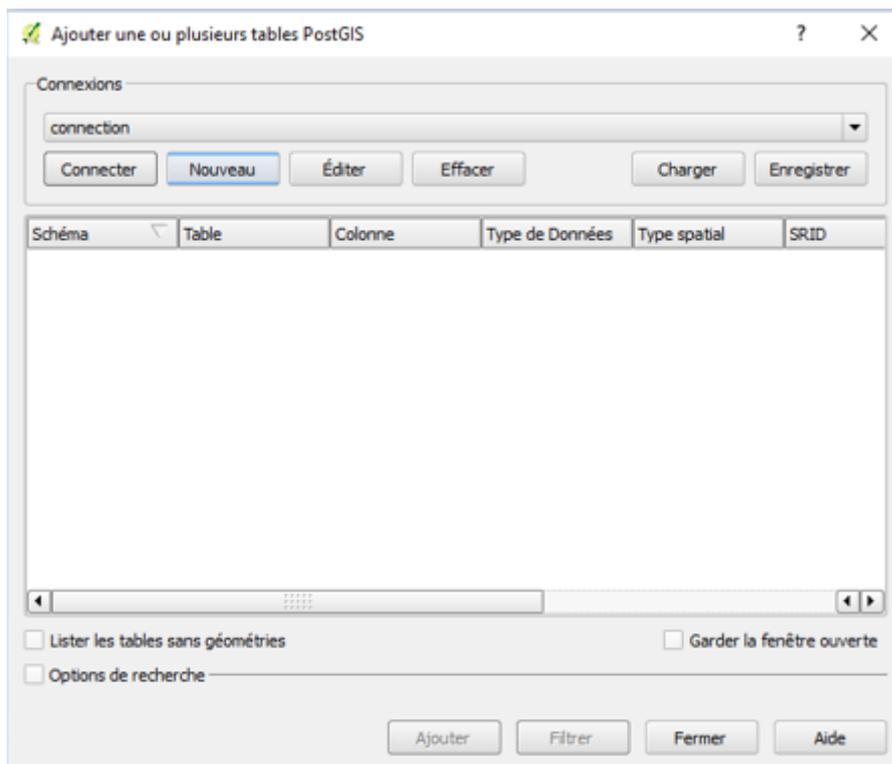


Figure 7 : Ajout d'une table PostGIS

La fenêtre « **Ajouter une ou plusieurs tables PostGIS** » apparaît et nous devons cliquer sur « **Nouveau** » pour accéder au gestionnaire de connexion.

Le tableau ci-dessous explique le paramètre de connexion PostGIS :

Paramètre	Description
Nom	Nom pour la connexion
Hôte	Nom pour l'hôte de la base de données. Il doit s'agir d'un nom existant, car il sera utilisé pour ouvrir une connexion Telnet ou interroger l'hôte. Si la base de données est sur le même ordinateur que QGIS, mettons simplement local host.
Base de données	Nom de la base de données.
Port	Le port par défaut est 5432.
Nom d'utilisateur	Nom d'utilisateur utilisé pour se connecter à la base de données.
Mot de passe	Mot de passe utilisé avec le Nom d'utilisateur pour se connecter à la base de données.
Mode SSL	Comment sera négociée la connexion SSL avec le serveur. Voici-les options : <ul style="list-style-type: none"> <li>– désactiver : essayer une connexion SSL non cryptée uniquement</li> <li>– permettre : essayer une connexion non-SSL. Si cela échoue, essayer une connexion SSL ;</li> <li>– préférer (par défaut) : essayer une connexion SSL. Si cela échoue une connexion non-SSL ;</li> <li>– requiert : essayer seulement une connexion SSL</li> </ul> Il faut noter qu'une accélération massive du rendu des couches Post GIS peut être obtenue en désactivant le SSL dans l'éditeur de connexion.

**Tableau 6** : Paramètre de connexion PostGIS

Après les paramètres et les options sont définis, nous pouvons tester la connexion en cliquant sur le bouton « **Tester la connexion** » afin d'assurer l'établissement de la connexion.

**PARTIE III :**  
**LA SOLUTION WEBMAPPING**

## CHAPITRE I. LE WEBMAPPING

### I- Notion sur le webmapping

#### 1. Définition

Le terme « *Webmapping* », largement utilisé sur internet, qui peut se traduire par cartographie en ligne est une partie du domaine de compétences des Systèmes d'Informations Géographiques (SIG). Un système d'information géographique est un système d'information permettant d'organiser et de présenter des données spatialement référencées permettant la production de plans et de cartes géographiques. Ses usages couvrent les activités géomatiques – ensemble des outils et méthodes permettant de représenter, d'analyser et d'intégrer des données géographiques - de traitement et de diffusion de l'information géographique. Le Webmapping est un domaine de compétences, des systèmes d'informations géographiques, permettant l'intégration et la visualisation de cartes géo-référencées sur internet depuis un navigateur web.

#### 2. Utilité

Le Webmapping permet de manipuler des cartes en ligne et de les intégrer à des sites. Pour que ces cartes soient les plus précises possible, les informations géographiques utilisées et manipulées par les services de cartographies en ligne sont collectées à un niveau international. Pour une meilleure interopérabilité de ces informations, les services de Webmapping obéissent à différentes normes imposées par l'*Open Geospatial Consortium* (OGC) qui est une organisation internationale qui développe et promeut des standards, afin de garantir l'interopérabilité des contenus, des services et des échanges dans les domaines de la géomatique et de l'information géographique.

#### 3. Contexte

La cartographie en ligne peut s'utiliser dans différents contextes. Depuis son émergence ces dix dernières années, le Webmapping est de plus en plus sollicité au quotidien aussi bien dans un contexte professionnel que personnel.

Dans le contexte professionnel, la dimension spatiale est devenue fondamentale dans la plupart des prises de décision avec l'apparition du Webmapping :

- Gestion des infrastructures et des équipements (développement, entretien et gestion des réseaux de collecte, de distribution, de communication...)
- Planification de l'occupation du sol : intégration et visualisation des contraintes.

- Réglementation et suivi : autorisation (d'exploitation, de prélever...) délivrées au regard des ressources de l'environnement.

Dans le contexte personnel, le webmapping s'est installé durablement dans les paysages de l'internaute :

- Recherche/visualisation d'un lieu ou d'un emplacement précis sur une carte en ligne.
- Calcul d'un itinéraire
- Géolocalisation d'une personne ou d'un objet

#### **4. Principes et fonctionnement du webmapping**

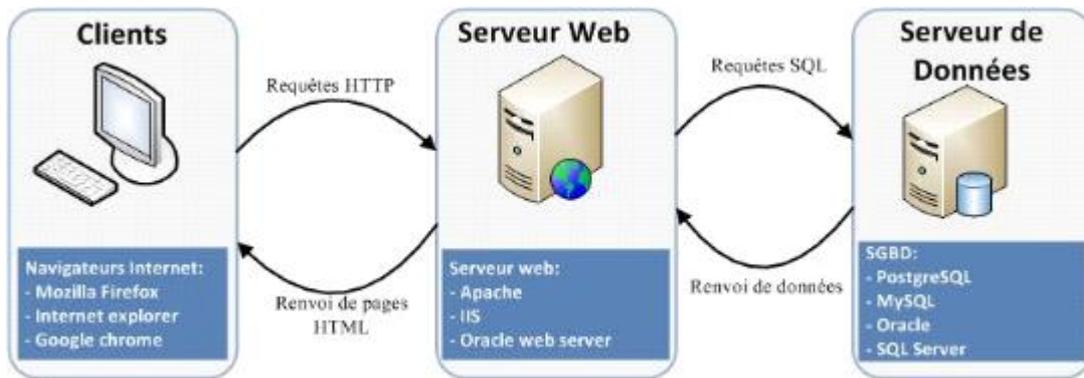
Comme défini précédemment, le WebSIG c'est la diffusion de données cartographiques par le biais de site web. Dès lors, l'accès à l'information devient dynamique : l'information est construite par un traitement déclenché à la demande de l'utilisateur. Ces technologies reposent sur celles du web que sont :

- architecture client/serveur ;
- protocole HTTP (*HyperText Transfer Protocol*) pour les échanges entre le client et le serveur ;
- stockage des données dans des bases de données interrogeables à l'aide du langage de requête SQL (*Structured Query Language*) ;
- les standards HTML, Javascript, AJAX, XML, SVG élaboré par le W3C (*World Wide Consortium*)

#### **5. Architecture d'une application web**

Le web c'est un ensemble de machines en réseau communiquant à l'aide d'un langage commun. Le web fonctionne en mode client/serveur c'est-à-dire qu'il y a des machines dites serveurs qui proposent des ressources et des machines appelées clients qui utilisent ces ressources. Les ressources sont par exemple des pages HTML, des images, des fichiers XML (*eXtensible Markup Language*) ou encore des programmes (PHP, Java, ASP.NET, Python, Perl, ...) chargés de les générer à la demande. Le client accède aux ressources à l'aide du protocole de communication HTTP.

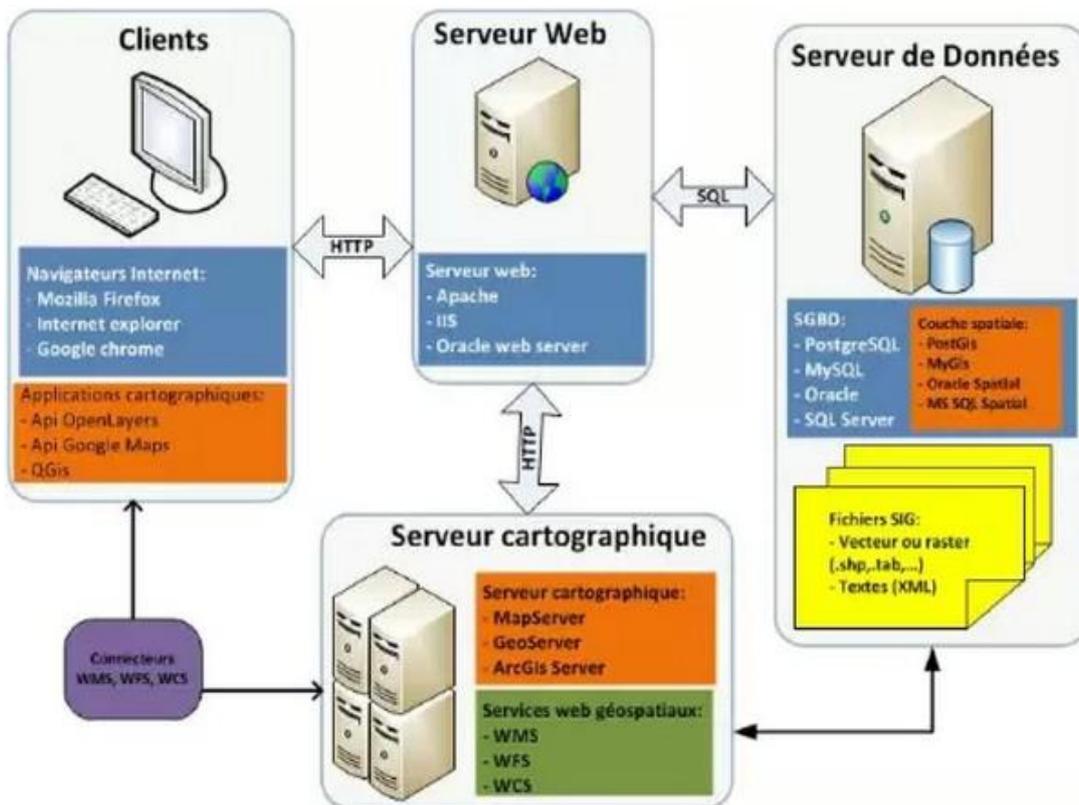
Au niveau des serveurs, en plus du serveur web, nous pouvons avoir un serveur de données qui va héberger le Système de Gestion de Base de Données (SGBD). Et pour y accéder, on utilise le langage universel d'interrogation des bases de données : SQL. Ci-dessous l'architecture d'une application web.



**Figure 8 :** Architecture d'une application web

## 6. Architecture d'une application webmapping

L'architecture d'une application de webmapping s'appuie sur celle du web. Ici, en plus des serveurs web et de données, nous avons le serveur cartographique. En fonction de la requête du client au serveur cartographique, il est retourné au client les données désirées sous la forme de carte. Ci-dessous l'architecture d'une application de webmapping.



**Figure 9 : Architecture d'une application de webmapping**

## **II- MISE EN ŒUVRE DE LA PLATE-FORME**

Cette partie du document est consacrée à la présentation des outils d'implémentation de la solution et à la description des activités menées pour la réalisation du système. Il faut dire que nous avons fait le choix des logiciels libres. Ainsi, nous avons utilisé PostgreSQL/postgis pour le système de gestion de la base de données, MapServer pour le serveur cartographique, Apache comme serveur Web et OpenLayers pour le client cartographique.

## **III- ENVIRONNEMENT ET OUTILS UTILISES**

### **1. Mapserver**

Issu de milieux universitaires (Université du MINNESOTA USA) et amélioré par des communautés de développeurs, MapServer est un serveur cartographique open source (à code ouvert) permettant de réaliser des applications de Webmapping. Le Webmapping étant à la fois le processus de génération des cartes ainsi que leur diffusion sur Internet et leur visualisation dans un navigateur web. Mapserver respecte les spécificités de l'OGC. Actuellement, la version 4.4 de Mapserver s'adapte quasiment à tout type d'environnement. Il peut être facilement étendu afin de supporter de nouveaux formats de données, environnements de développement, systèmes d'exploitation ou serveurs Web. En entrée, il accepte une multitude de formats de données géographiques. En sortie, il produit des cartes interactives à destination d'Internet sur plusieurs formats.

### **2. Mapserver et Apache**

Apache est un programme permettant d'implémenter un ordinateur en serveur Web. Il utilise le module PHP pour interpréter les scripts. Il est installé directement lors de l'installation de Mapserver. C'est donc un programme capable d'interpréter les requêtes

HTTP arrivant sur le port associé au protocole HTTP (par défaut le port 80), et de fournir une réponse avec ce même protocole. Mapserver peut être utilisé en CGI. Il faut rappeler qu'un serveur web est un logiciel permettant à des clients d'accéder à des pages web, c'est-à-dire des fichiers au format HTML à partir d'un navigateur (aussi appelé browser) installé sur leur ordinateur. Un des principaux intérêts de l'utilisation de CGI est la possibilité de fournir des pages dynamiques.

#### **2.1. Caractéristiques**

MapServer est performant en termes de vitesse d'affichage des cartes.

Il est très fiable car il peut faire face à plus de 150000 connections simultanées.

En termes d'adaptabilité et d'évolutivité, il s'accommode quasiment à tous types d'environnement. Il peut être facilement étendu afin de supporter de nouveaux formats de données, environnement de développement, système d'exploitation ou serveur web.

Il peut facilement intégrer différents types d'éléments cartographiques dans une application tel que l'échelle, la légende, la visibilité des couches dépendant de l'échelle, système de prévisualisation sophistiqué.

## 2.2. Fonctionnalité

Mapserver est installé sur la machine serveur.

Il est capable de stocker et gérer des données localisées. Le stockage des données attributaires passe par un Système de Gestion de Bases de Données Relationnelles.

Il permet de mettre à jour des données graphiques et attributaires.

Le serveur de carte doit être couplé avec un SGBDR capable de gérer la spatialité.

Il retourne les informations sous forme adaptée aux navigateurs Web ;

Il doit permettre le croisement de données, le calcul et la mesure de longueur et de superficie.

## 2.3. Avantages

Le principal avantage de MapServer est le prix car il est totalement gratuit.

Incorpore plusieurs langages de programmation.

Il Fonctionne avec un maximum d'explorateur client.

Il Utilise des formats ouverts ou des formats propriétaires (shapefile, geotiff, tab).

## 2.4. Inconvénient

Comme la plupart des logiciels Open source :

Mapserver nécessite un personnel motivé, et formé ;

Il présente les difficultés d'installation ;

Il nécessite un temps d'investissement humain important.

### 3. PostgreSQL

Notre sujet s'inscrit dans un cadre Open source, c'est-à-dire logiciels dont les sources sont accessibles et modifiables. Dans ce contexte, plusieurs SGBD nous sont proposés. Les plus connus sont MySQL et PostgreSQL. D'une manière générale PostgreSQL est un Système de Gestion de Base de Données Relationnelles (SGBDR) développé au département d'Informatique de l'université de Californie et fonctionnant sur des systèmes de type UNIX ou WINDOWS. Son architecture est de type client/serveur. Il est ainsi constitué d'une partie serveur, dont le programme est postmaster, traitant les requêtes des clients et d'une partie client permettant d'accéder aux données. PostgreSQL supporte une grande partie du standard SQL tout en offrant de nombreuses fonctionnalités modernes :

- requêtes complexes ;
- clés étrangères ;
- triggers ;
- vues ;
- intégrité transactionnelle ;
- contrôle des versions concurrentes (MVCC ou *multiversion concurrency control*).

De plus, PostgreSQL apporte une puissance additionnelle substantielle en incorporant les quatre concepts de base ci-après afin que les utilisateurs puissent facilement étendre le système. Il s'agit des concepts de classes, héritage, types, fonctions.

D'autres fonctionnalités accroissent la puissance et la souplesse : Ce sont les méthodes d'indexation, opérateurs, contraintes et les fonctions d'agrégat.

Ces fonctionnalités placent PostgreSQL dans la catégorie des bases de données objets relationnels. Ainsi, bien que PostgreSQL possède certaines fonctionnalités orientées objets, il appartient avant tout au monde des SGBDR.

#### 3.1. Vocabulaire

Client/serveur : c'est une architecture dans laquelle plusieurs postes de travail dépendent d'un poste contenant toutes les ressources du système.

Le terme site désigne l'hôte sur lequel PostgreSQL est installé.

Le super utilisateur (super user) Postgres est l'utilisateur propriétaire des binaires (programmes) et des fichiers de la base de données.

L'administrateur des bases ou DBA est la personne responsable de l'installation de Postgres et de la politique de sécurité associée. Le DBA peut ajouter de nouveaux utilisateurs et créer des bases.

Le postmaster est un processus qui reçoit et centralise les requêtes expédiées au système PostgreSQL. Il peut également être défini comme un serveur de bases de données multi-utilisateurs de PostgreSQL.

### 3.2. Caractéristiques de PostgreSQL

PostgreSQL possède de nombreuses caractéristiques faisant de lui un SGBDR robuste et puissant digne des SGBDR commerciaux. PostgreSQL dispose :

Interfaces graphiques Windows et DOS nécessaire pour gérer les bases de données ;

Bibliothèques écrivent en plusieurs langages ;

PostgreSQL peut être employé comme base de données spatiale principale pour les Systèmes d'Information Géographique.

Une API ODBC permettant à n'importe quelle application supportant ce type d'interface d'accéder à des bases de données de type PostgreSQL.

### 3.3. Fonctionnement de PostgreSQL

PostgreSQL fonctionne selon une architecture client/serveur. Pour qu'une application cliente accède à une base de données, elle se connecte via le réseau ou localement à un postmaster en cours d'exécution. Ensuite, le postmaster déclenche un processus serveur séparé pour gérer la connexion. Lorsqu'une requête est faite, le processus Postmaster lance un nouveau processus fils appelé postgres qui va établir la connexion entre le client et le serveur PostgreSQL. Une fois la connexion établie, le processus client peut envoyer une requête au serveur. La requête est transmise en texte simple, c'est-à-dire qu'aucune analyse n'est réalisée au niveau de l'interface client. Le serveur analyse la requête, crée un plan d'exécution, exécute le plan et renvoie les lignes trouvées au client par la connexion établie. Ainsi PostgreSQL peut gérer plusieurs connexions à partir des processus fils qu'il génère.

## 4. PostGIS

PostGIS est un module d'extension de PostgreSQL, permettant offrant des fonctionnalités pour les objets géoreférencés et géométriques. PostGIS est développé par Réfractions Research inc., comme projet de recherche spatial de technologie de base de données. Réfractions Research inc. est une compagnie de consultation de base de données se spécialisant dans l'intégration de données et les logiciels personnalisés de développement. PostGIS est une prolongation du système de gestion de base de données PostgreSQL.

PostGIS utilise deux bibliothèques principales GEOS et Proj4. Geos est une bibliothèque qui enrichit ou complète la panoplie des fonctions spatiales de PostGIS tandis que Proj4 est une bibliothèque permettant la

re-projection dans les divers systèmes de projection connus. Un système de projection étant une transcription sur un plan d'une surface courbe sans trop altérer ou provoquer des déformations.

#### 4.1. Caractéristiques

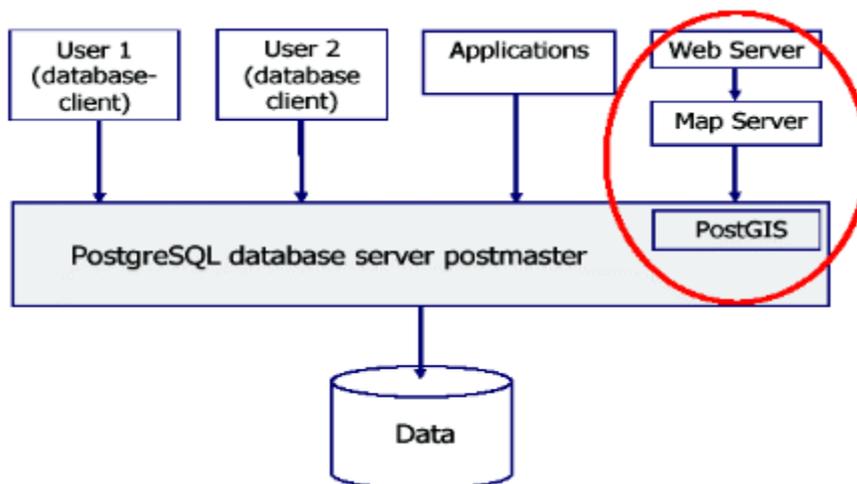
PostGIS est implémenté conformément aux spécifications SQL Standard de l'OGC ;

PostGIS est actuellement à sa version 0.8.0, couplable avec GEOS lui apportant des fonctionnalités bien utiles telles que `Within()`, `Disjoint()`, `Touche()`, `GeomUnion()`, `Intersection()`, `Buffer()`...

PostGIS permet de créer des bases de données afin de stocker et de traiter les données géométriques ;

PostGIS interagit avec plusieurs autres langages tel que le C, le PHP, la java ...

L'architecture des applications de Webmapping est illustrée par le schéma suivant.



**Figure 10 :** Mise en évidence des applications Web et PostGIS/PostgreSQL

## **IV- Langages utilisés**

Nous avons utilisé les langages, HTML, JavaScript, PHP pour la réalisation de notre application.

### **1. HTML**

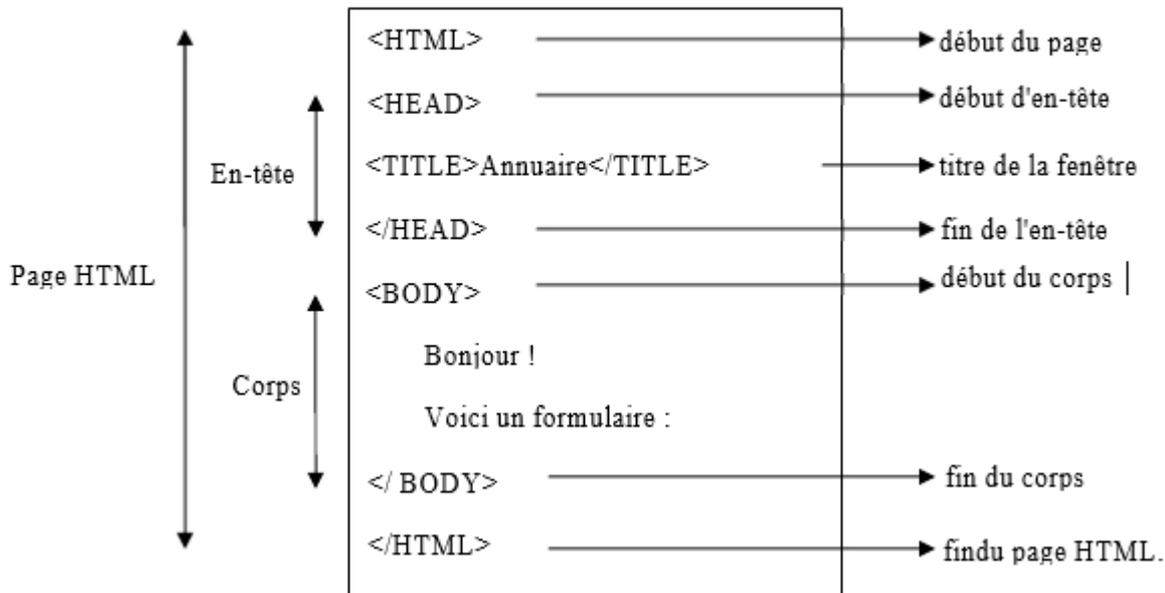
L'HTML est un langage de description de document multimédia utilisé par le Web. Il utilise des balises ou tags pour indiquer la façon dont le document doit être affiché. Les balises sont délimitées par les signes '<' et '>'. C'est le langage universel utilisé pour communiquer sur le Web. Les informations seront ainsi transportées sur cette gigantesque toile de réseaux interconnectés qui est l'Internet, pour aboutir sur l'ordinateur du lecteur grâce à un programme appelé navigateur ou browser. HTML (HyperText Markup Language) est le langage de base pour concevoir des pages destinées à être publiées sur le Web. Il permet la mise en forme du contenu d'une page web.

#### **a- Le document HTML**

Un document HTML est un document envoyé généralement par un serveur WWW vers un poste client équipé d'un navigateur comme Mozilla Firefox, Opéra, Internet Explorer permettant de visualiser le document. Un document HTML contient des informations multimédias : textes, images fixes, sons, vidéos. En fait, il est un fichier spécial dont le suffixe est .htm où .html.

#### **b- Physionomie d'une page HTML**

La structure de base d'une page HTML est le suivant



**Figure 11 :** Interface d'une page HTML

## 2. JavaScript

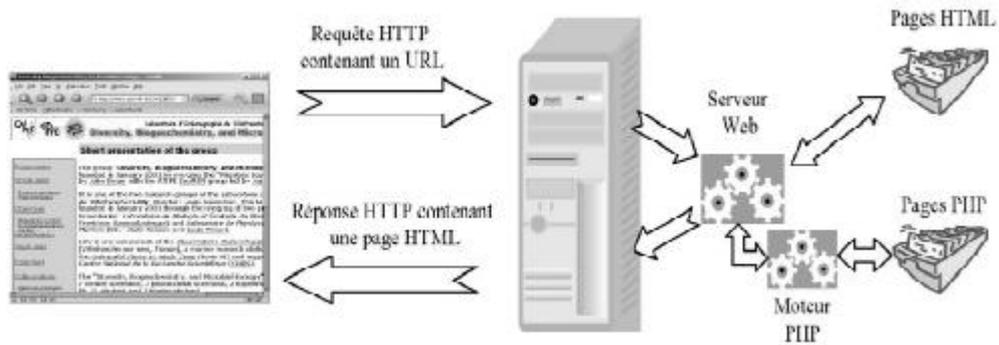
Le **JavaScript**, langage de script incorporé dans un document HTML. Il est principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe.

## 3. PHP

PHP (*Hypertext Preprocessor*) est un langage interprété (un langage de script) et exécuté du côté serveur. PHP est l'un des langages les plus utilisés dans le développement web et mieux depuis la prise en charge dans sa version 4 de la Programmation Orientée Objet (*POO*). Il dispose de près de 3 000 fonctions utilisables dans des applications très variées et couvre pratiquement tous les domaines en rapport avec les applications web. Presque tous les SGBD du marché peuvent s'interfacer avec PHP (commerciaux ou venant du monde libre). Nous l'avons utilisé pour insérer des données dans la base de données à partir des formulaires de saisie aussi pour extraire des données de la base de données.

### a- Place du PHP sur le serveur Web

L'emplacement du PHP dans un serveur Web est représenté dans la figure suivante



**Figure 12 :** Place du PHP sur le serveur WEB

## CHAPITRE II. REALISATIONS

L'un des objectifs du projet est de réaliser un prototype de la plate-forme. Pour aboutir à ce prototype, ce chapitre comprend les sections suivantes :

- Mise en place et configuration du serveur cartographique ;
- Création de fichier Mapfile
- Changement de Mapfile
- Test du serveur
- Élaboration des interfaces de cartographie et d'administration du système.

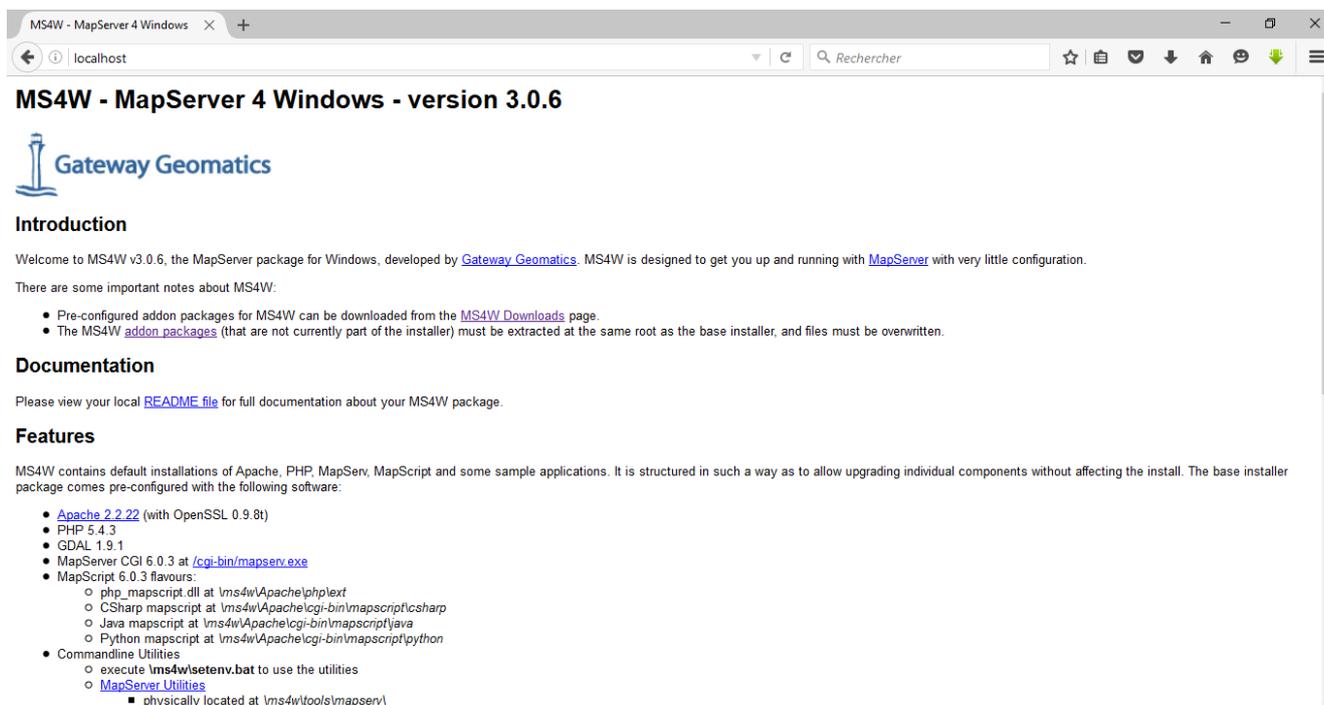
### I- Mise en place du serveur cartographique avec Mapserver

L'installation de MapServer sur un système Windows peut se faire avec le paquet MS4W (*MapServer for Windows*) ou OSGeo4W (*OSGeo for Windows*). Nous avons utilisé MS4W pour notre installation. MS4W est un paquetage gratuit qui regroupe les binaires précompilés de l'ensemble des composants nécessaires pour installer un service web basé sur le serveur Apache, le langage PHP ainsi que MapServer. Il est très simple à installer.

#### Etapas d'installation :

- décompression du paquet sous la racine C:/, nous obtenons C:/ms4w qui contient les fichiers d'installation ;
- lancement du serveur apache avec la ligne de commande « *apache-install.bat* » sur DOS en se plaçant sur C:/ms4w ;
- configuration du serveur Apache.

Nous obtenons la page de démarrage de MapServer ci-dessous en tapant dans notre navigateur l'adresse *:localhost* (Figure 13).



**Figure 13 :** Fenêtre de démarrage de MapServer

A présent, notre serveur cartographique MapServer est bien installé et fonctionnel.

## 1. Création et configuration du mapfile

Divers moyens existent pour créer un mapfile. Un mapfile peut être créé à partir de logiciels dédiés logiciel QGIS<sup>22</sup> (Quantum Gis) ou en éditant un fichier avec un éditeur de texte tel Notepad+. Dans notre cas nous avons créé notre mapfile avec Notepad+. QGIS est un logiciel SIG complet OpenSource. Il prend en charge de nombreux formats vectoriels, rasters ainsi que les formats et fonctionnalités de plusieurs bases de données. Il permet la gestion de tables spatiales PostGis, de formats shapefiles, couvertures ArcInfo, Mapinfo, et autres fichier supportés par OGR<sup>23</sup>; des rasters dans un grand nombre de formats; l'identification, la sélection et l'étiquetage, la numérisation d'objets graphiques, l'affichage des tables attributaires, etc. QGIS supporte également le standard WMS de l'OGC.

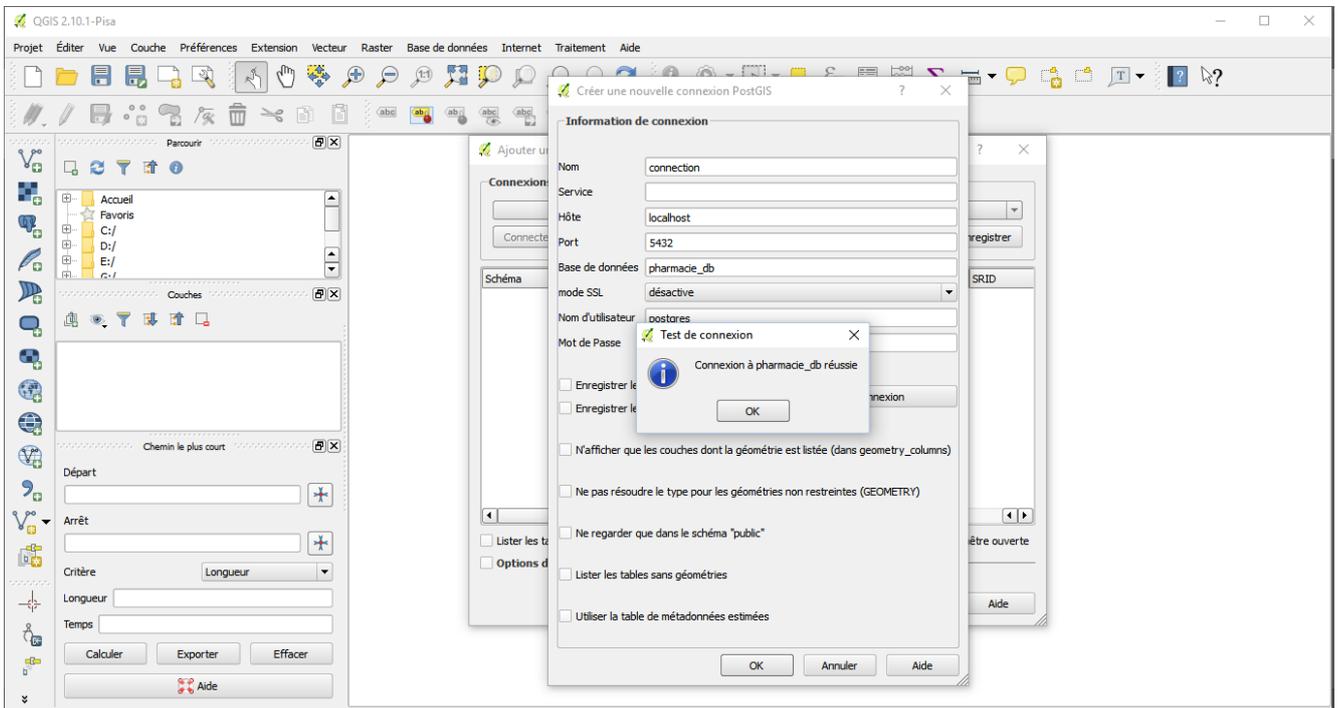
Après avoir installé QGIS, nous avons suivi les étapes suivantes pour la création de notre mapfile.

### Création de projet sous QGIS

Il a consisté à créer un projet sous QGIS et de le nommer.

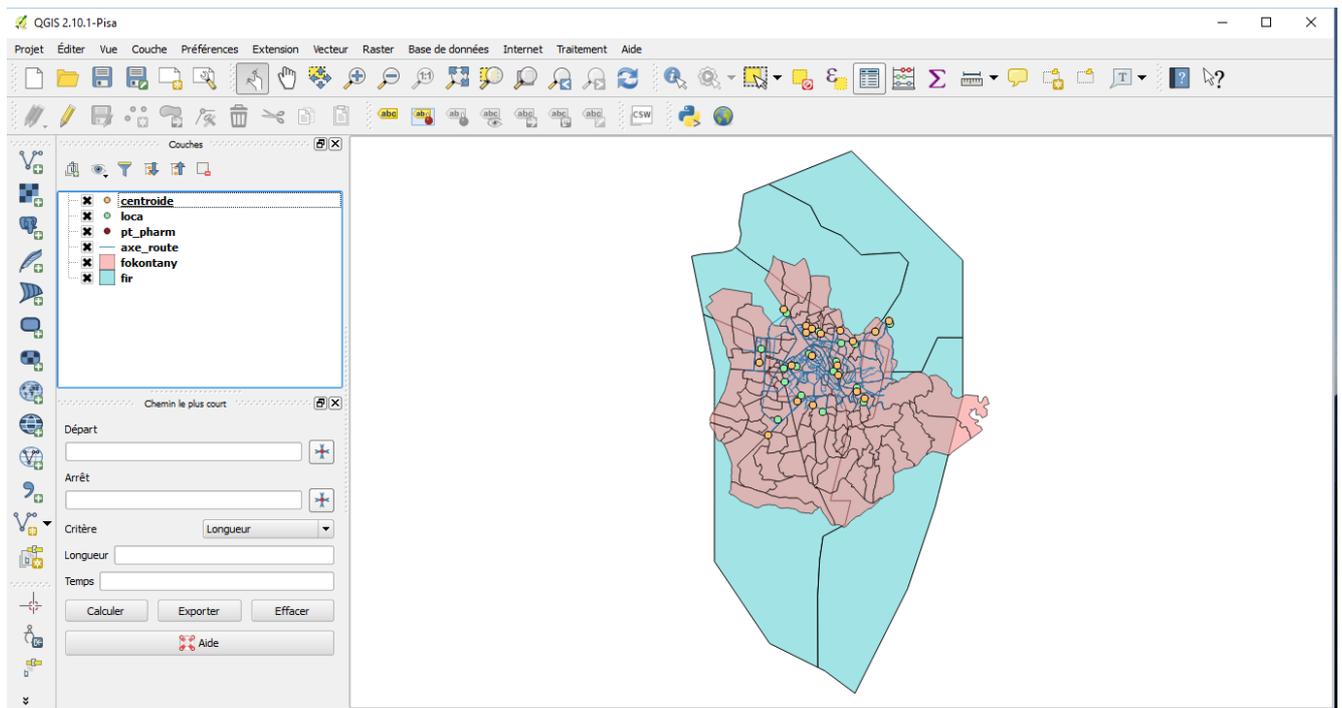
### Ajout de couche PostGis au projet QGIS

Il s'agit d'importer dans QGIS les données de notre table qui a une colonne géométrique sous PostgreSQL. Une connexion a été établie entre QGIS et Postgis, ci-dessous la fenêtre de connexion (Figure 14).



**Figure 14 :** Fenêtre de connexion à Postgis

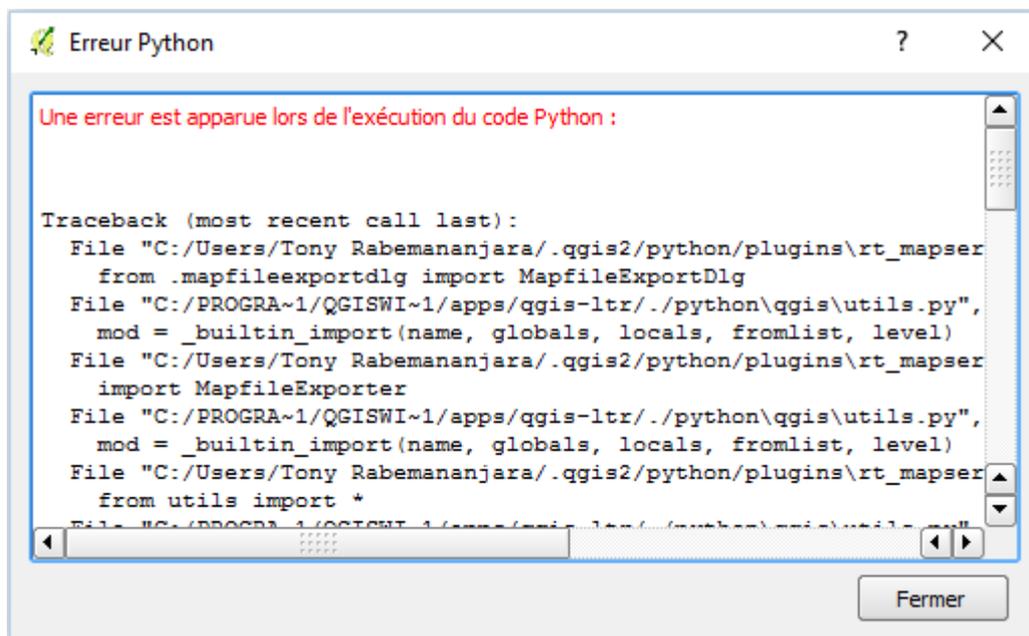
Une fois la connexion établie, nous importons les couches de notre base de données PostgreSQL. (Figure 15).



**Figure 15 :** Vue de la couche sous QGis

### Export vers MapServer

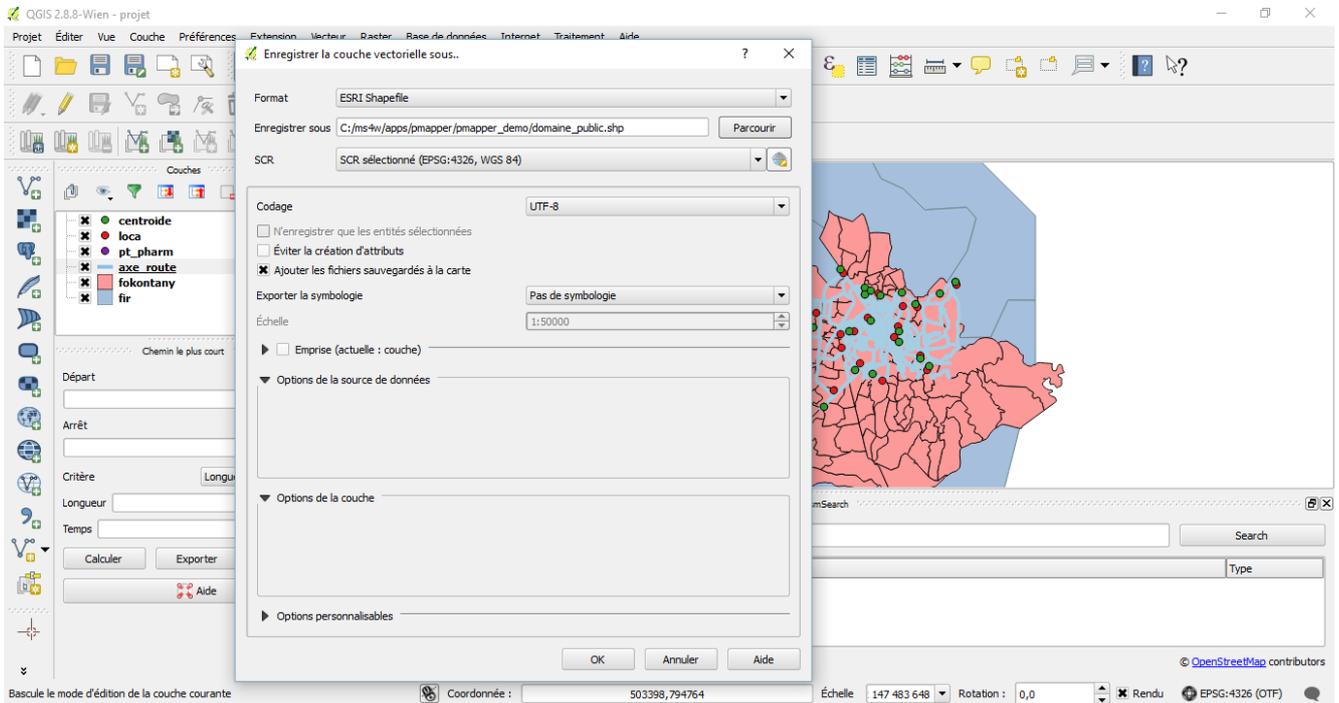
L'export de notre couche vers MapServer consiste à générer un mapfile. Sur QGIS, si on passe par l'extension « RT Mapserver Exporter », on a ce message d'erreur qui apparaît (Figure 16).



**Figure 16 :** Erreur Python

Mais pour pouvoir exporter cette couche sur Mapserver, on va l'exporter sur PMAPPER : On Clique droite sur la couche vue/Enregistrer sous. La fenêtre ci-dessous apparaît

Ci-dessous la fenêtre d'export vers PMAPPER (Figure 17).



**Figure 17** : Export de la couche vers PMAPPER

Il ne nous reste plus qu'à éditer le mapfile généré répondant à nos besoins.

### Configuration de notre mapfile généré

Le mapfile est le fichier de configuration de MapServer. C'est un fichier texte qui va contenir tous les paramètres nécessaires à MapServer pour la génération d'un document cartographique, statique ou dynamique. Il est composé d'une hiérarchie d'objets où chaque objet peut contenir d'autres objets et/ou des propriétés.

Notre mapfile ainsi généré, nous avons procédé à sa configuration (Annexe 3). Ainsi, nous avons :

- créé un service WMS ;
- définis les paramètres de connexion à PostgreSQL de nos données ;
- spécifié les projections des couches ;

## 2. Développement de l'interface cartographique

L'interface cartographique est la fenêtre qui va contenir notre carte. Son développement a été réalisé grâce à l'utilisation du PMAPPER. Cette interface cartographique contient la carte et les outils de manipulation. La carte est composée de fond de carte (couche de base) et de couches secondaires. Nous avons utilisé les fonds de carte ortho

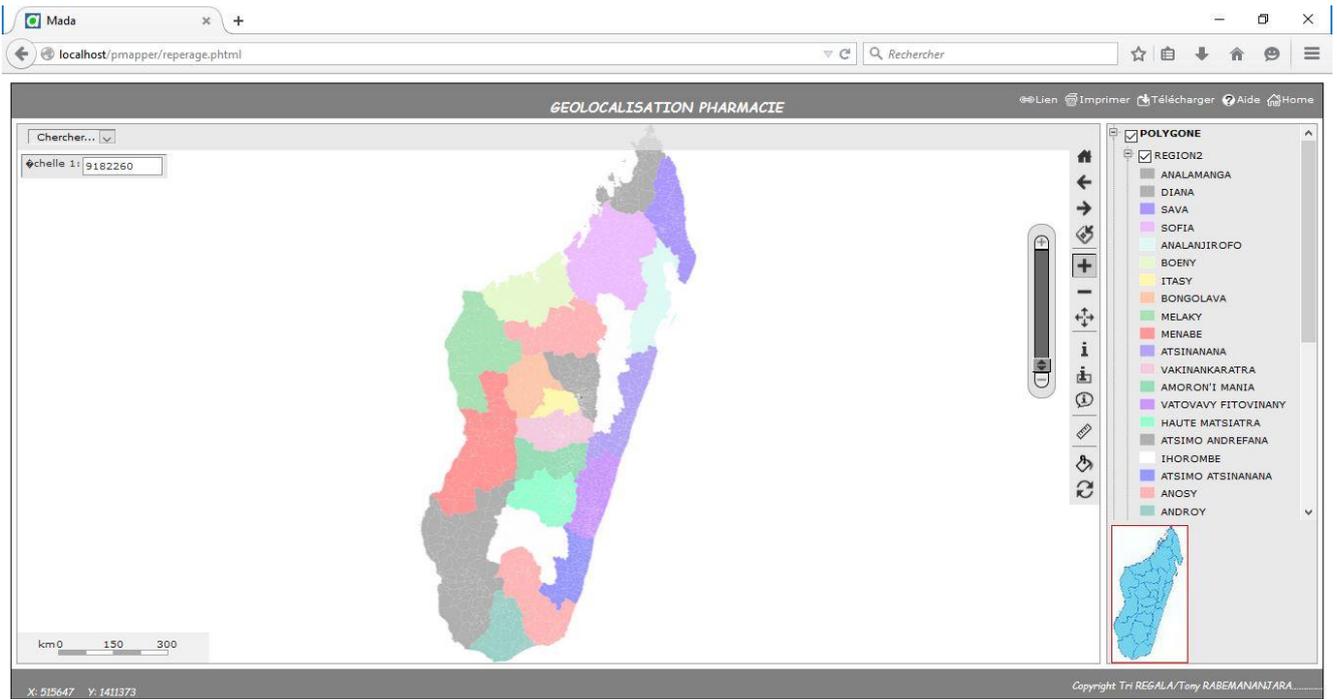
### 2.1 Utilisation des Fonds de carte

#### Fond de carte Ortho

Ortho est un service de cartographie en ligne qui met à disposition des fonds de cartes utilisables par défaut. Ces fonds de cartes sont produits par Google à partir de sources de données telles que Télé Atlas<sup>24</sup>©, Navteq<sup>25</sup>©, IGN, [Spot Image](#), etc. Toutes les régions du globe n'ont pas la même précision en termes de couverture des données cela par manque d'intérêt, les difficultés d'accès et la législation.

L'ortho propose plusieurs vues dont le plan classique, le relief, le satellite. Pour intégrer le fonds de carte Ortho, nous avons fait appel à Ortho dans un fichier mapfile et nous avons initialisé la carte dans le fichier.

## Ouvrons le navigateur web



**Figure 18 : Aperçus dans PMAPPER**

## CONCLUSION

L'étude a démarré par la définition d'une méthodologie de travail qui nous a conduits tout au long du projet. Ensuite, il y a la mise en place des données, la base de données fut construite sous le serveur PostgreSQL. Son extension spatiale Postgis fut utilisée pour le stockage des données à caractères géographiques.

L'architecture d'un webmapping emploie celle du web donc utilise un serveur web en plus de serveurs cartographiques. Apache a été configuré et utilisé pour jouer le rôle de serveur web et nous avons fait appel à MapServer comme serveur cartographique afin de diffuser les données géographiques étant donné sa robustesse.

Tout au long du projet, nous avons dû côtoyer différents langages de programmation tels que HTML, PHP, Javascript..., nous avons également rencontré quelques difficultés avec certaines fonctions Javascript mais tout a été résolu et réalisé avec succès. Cela nous a permis d'enrichir et consolider nos connaissances en programmation orientée web.

A l'issue de notre étude, nous avons pu établir une base de données où figure la localisation des pharmacies à Antananarivo grâce au système informatisé du Webmapping.

Cet outil répond au besoin des utilisateurs, facilite la vie quotidienne et peut même sauver une vie.

On peut envisager d'appliquer ce base de données dans les autres villes et dans d'autres domaines tel la localisation d'un hôpital, d'un marché public ...

Enfin, pour la prochaine étape, on pourra étendre ce projet en l'appliquant dans un smartphone possédant un système android.

## GLOSSAIRE

**API : Application Programming Interface** ou Interface de Programmation est un ensemble de fonctions, procédures ou classes mises à disposition par une bibliothèque logicielle, un système d'exploitation ou un service. Elle permet l'interaction des programmes les uns avec les autres.

**Logiciel Libre** : définition : Logiciel que tout un chacun a la liberté d'exécuter, de copier, distribuer, étudier, modifier et améliorer.

**OGC : Open Geospatial Consortium** : L'OGC est une organisation internationale à but non lucratif qui a pris la tête dans l'élaboration de standards dans le domaine géospatial.

**SIG : Un Système d'Information Géographique** est un système d'information permettant d'organiser et de présenter des données alphanumériques spatialement référencées, ainsi que de produire des plans et des cartes. Ses usages couvrent les activités géomatiques de traitement et diffusion de l'information géographique.

**URL : Uniform Resource Locator** : Chaîne de caractères utilisée pour identifier les ressources dans le World Wide Web.

**WFS : Web Feature Service** : Interface web autorisant des manipulations sur des objets géographiques. Les données sont transmises au format GML (standard XML).

**WMS : Web Map Service** : Le **Web Map Service** est une norme définie par l'OGC permettant au moyen d'une URL formatée d'interroger un serveur cartographique afin d'afficher une carte/image.

**GML : Geography Markup Language** : Format standardisé pour le transport de l'information géographique.

**GPL : General Public Licence** : License d'utilisation des logiciels assurant la libre et gratuite utilisation tant du logiciel que des sources.

**XML : eXtensible Markup Language** : standard du World Wide Web Consortium qui sert de base pour créer des langages balisés spécialisés.

## Bibliographie

- [1] Regala Trinome Joseph, Mémoire de fin d'étude : « Contribution à la mise en place d'une base de données de géolocalisation des pharmacies de la ville d'Antananarivo »
- [2] P. Roques et Franck Vallée, UML 2 en action, de l'analyse des besoins à la conception, Editions Eyrolles, 4e Edition
- [3] P. Roques, UML2 Modéliser une application Web, Editions Eyrolles, 4e Edition, 2008.
- [4] F. Goblet, Michel Dirix, L Goblet et J Philippe Moreux, Développer avec les API Google Maps, Dunod, 2010
- [5] Tyler Mitchell, Web Mapping illustrated, O Reilly, 2005
- [6] Laurent Jégou, Cartographie et SIG interactif en lignes, Scéance1 : le serveur web SIG, principe et solution, Atelier cartographique 2010.
- [7] Magalie BRUNET, Adrien SAUNAL, Evelyne REYREAU, Réalisation d'un prototype sur une architecture client/serveur pour la saisie des données géographique à distance, SIGMA.
- [8] OUEDRAOGO Jean Claude, Mémoire MIASIG 2010-2011 : Mise en place d'une interface webmapping sur la « Capitalisation des expériences de gestion durable de la fertilité des sols au Burkina Faso », Novembre 2011

## Webographie

### Consulté le :

<http://mapserver.org/fr/documentation.html>

<http://docs.postgresql.fr/9.2/>

<http://www.postgis.fr/chrome/site/docs/workshop-foss4g/doc/index.html>

[http://www.postgis.fr/chrome/site/docs/workshop-routing-foss4g/docs/\\_build/html/index.html](http://www.postgis.fr/chrome/site/docs/workshop-routing-foss4g/docs/_build/html/index.html)

<http://geotribu.net/node/>

<http://docs.openlayers.org/>

<http://trac.osgeo.org/openlayers/wiki/HowToDownload>

<http://www.geoext.org/tutorials/index.html>

[http://download.osgeo.org/qgis/doc/manual/qgis-1.7.2\\_user\\_guide\\_fr.pdf](http://download.osgeo.org/qgis/doc/manual/qgis-1.7.2_user_guide_fr.pdf)

<http://dl.maptools.org/dl/>

<http://www.mapserver.org/mapfile/>

<http://qgis.org>

# ANNEXES

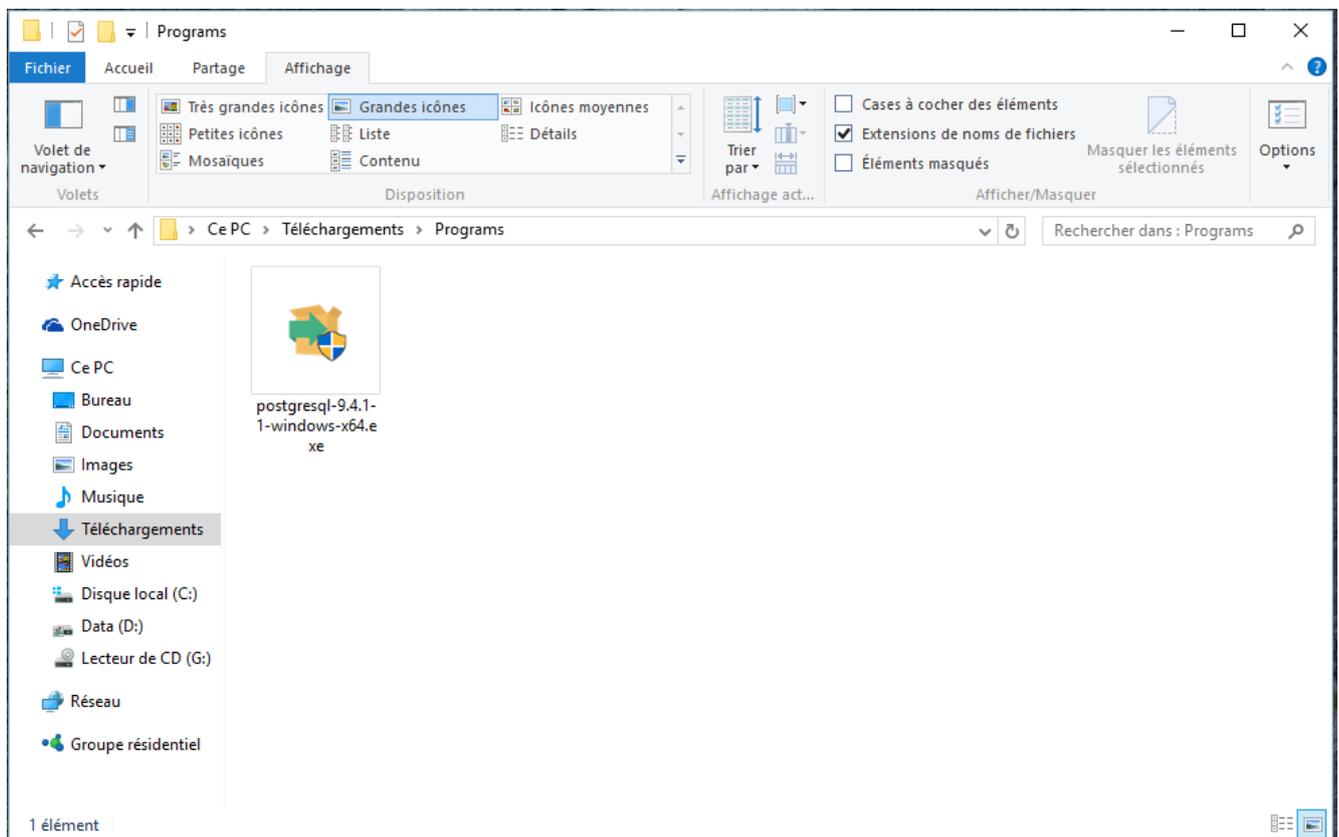
## Annexe 1 : Installation de PostgreSql/PostGIS

PostgreSQL est un logiciel libre, pour l'obtenir il suffit de le télécharger dans la source qui est disponible sur le lien : <http://www.master.postgresql.org/download/mirrors-ftp>.

Il existe plusieurs versions disponibles pour le logiciel mais nous avons utilisé la version PostgreSQL 9.4 que nous trouverons le plus stable et à jour.

Le fichier téléchargé est un fichier d'archive « .zip » que nous stockons dans le répertoire PostgreSQL de notre disque dur.

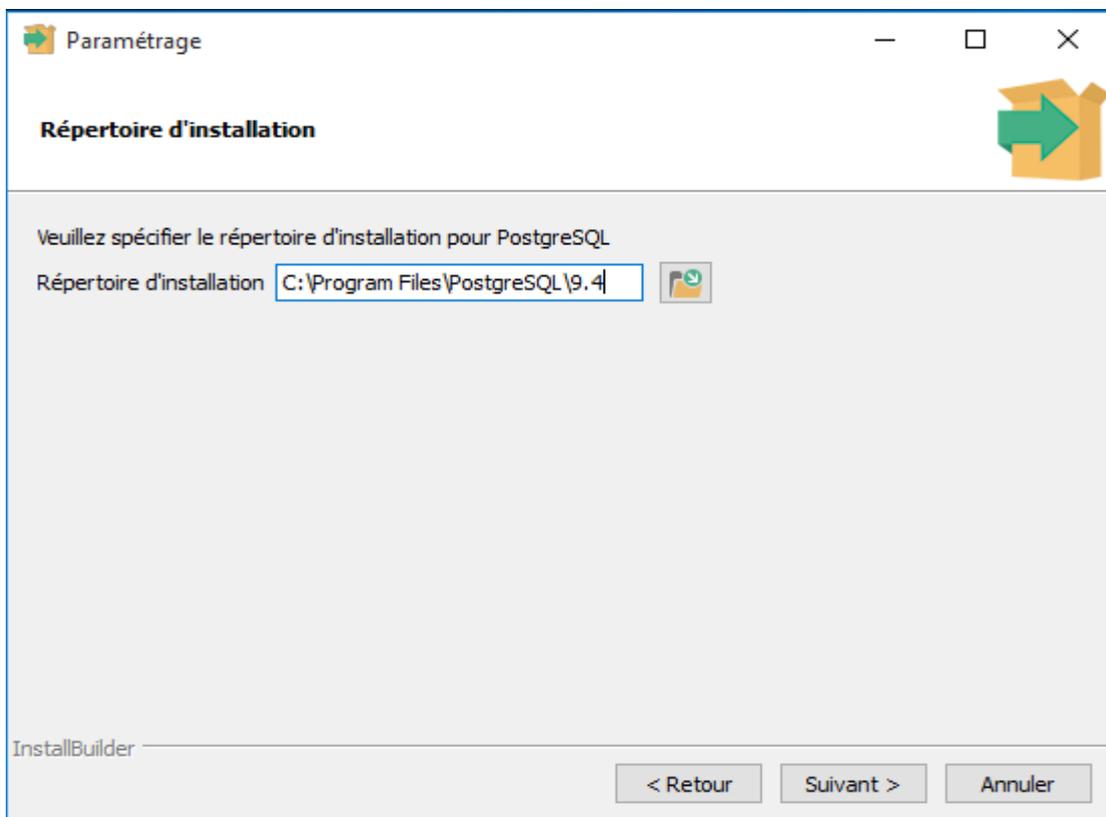
Après extraction du fichier compressé, nous lançons l'application en double cliquant sur le fichier exécutable

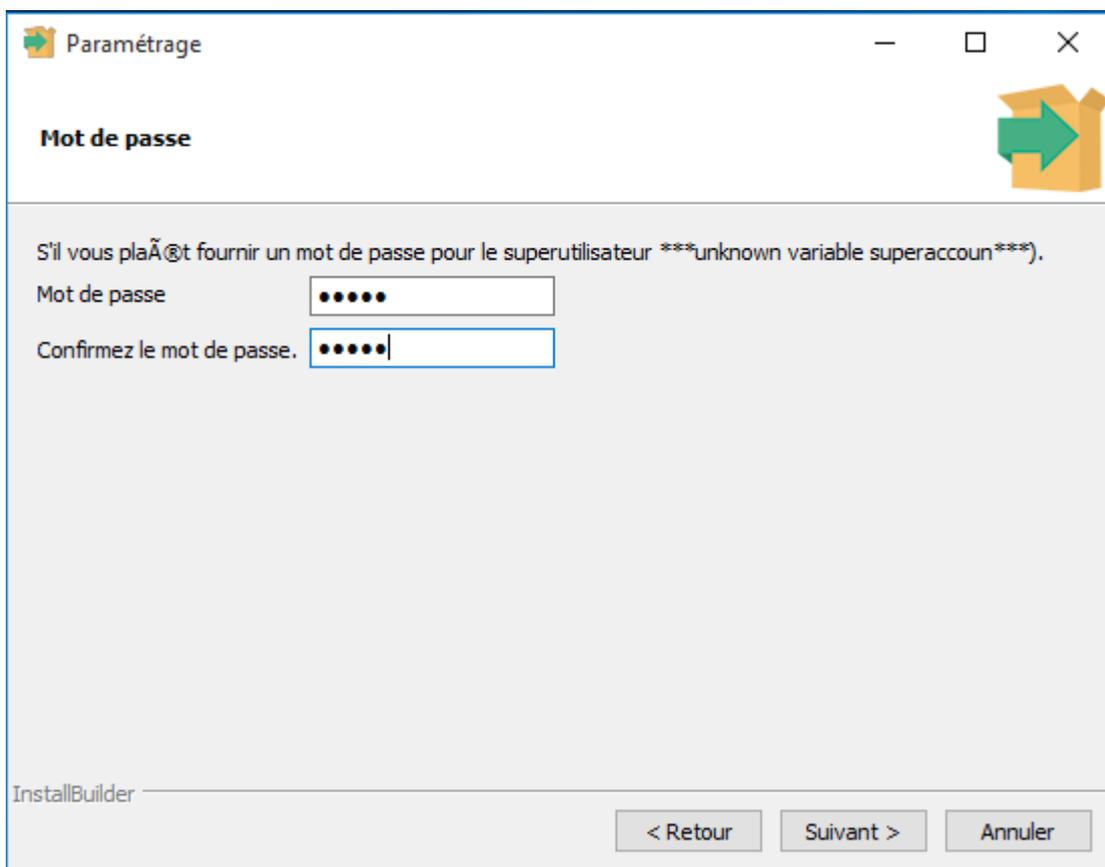
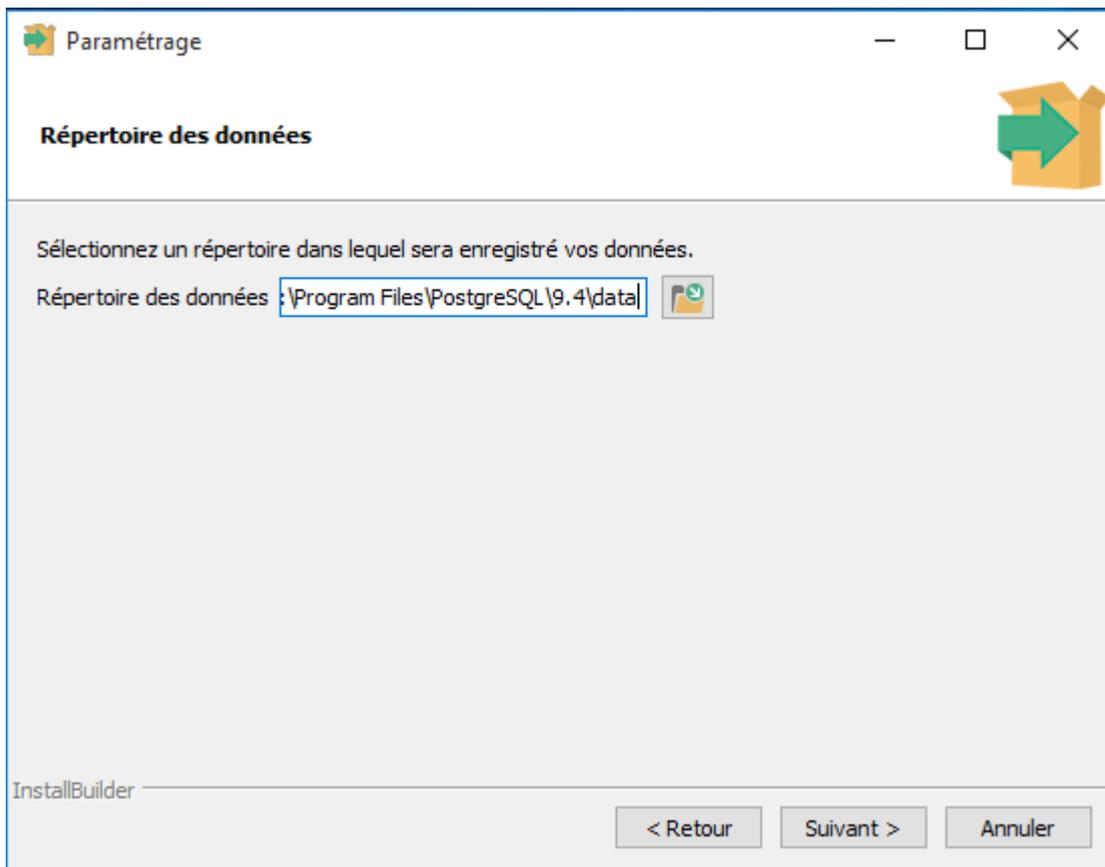


Ainsi, une fenêtre de bienvenue s'ouvre pour indiquer le début d'installation. On clique tout simplement sur « **suivant** » pour continuer l'installation.

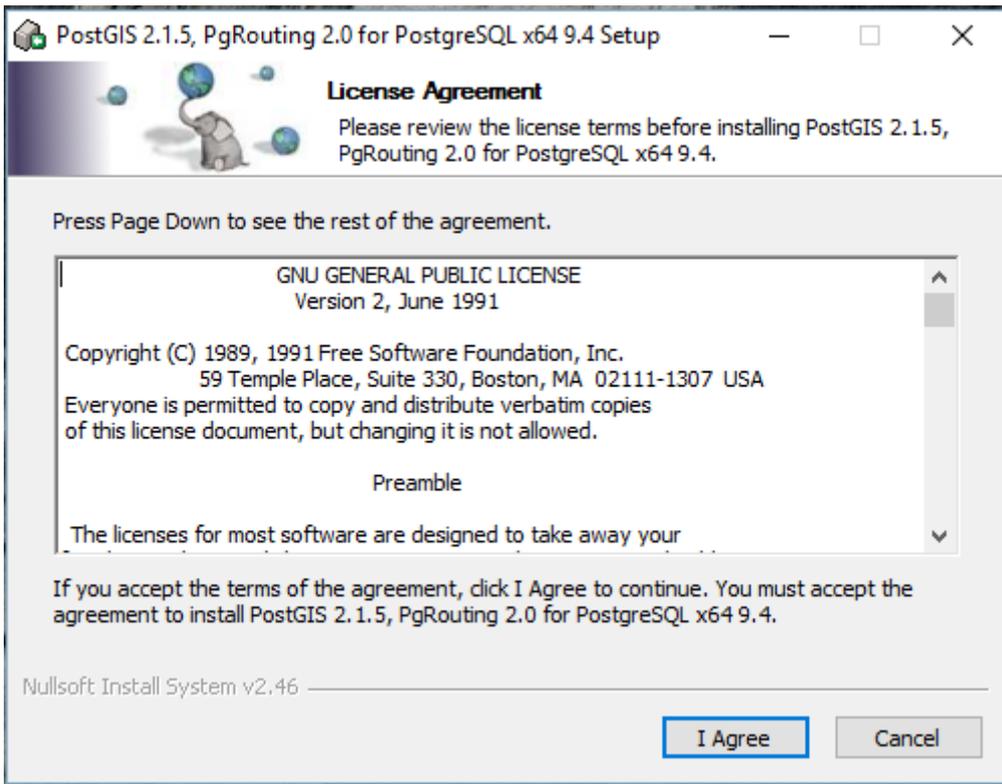


Choisir le répertoire

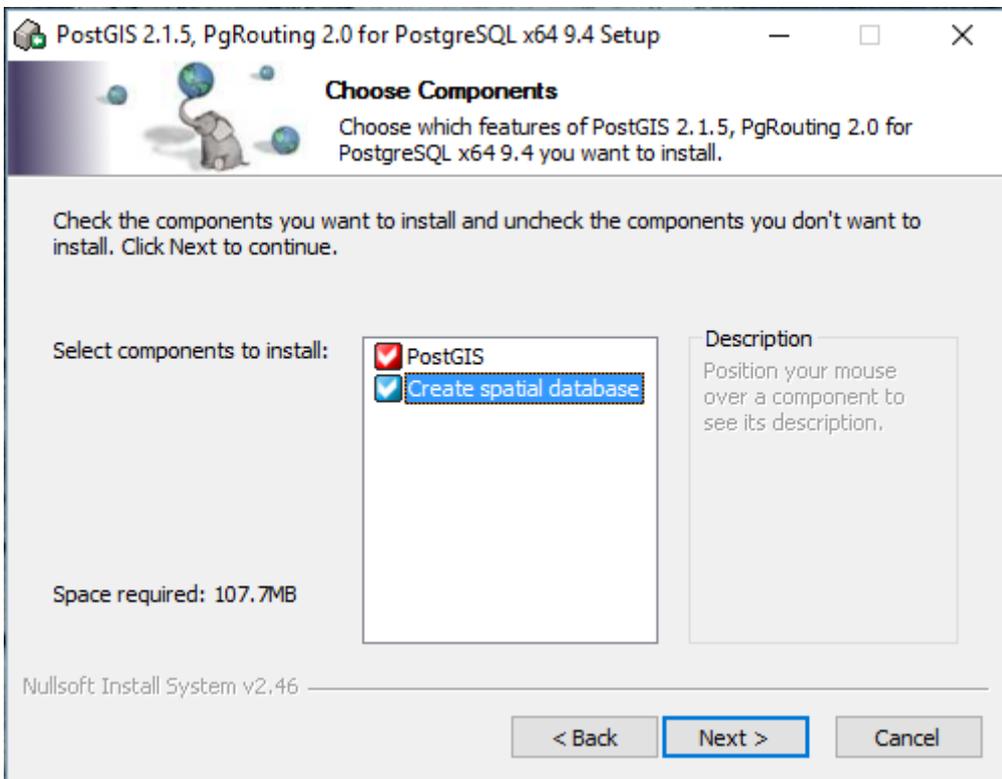




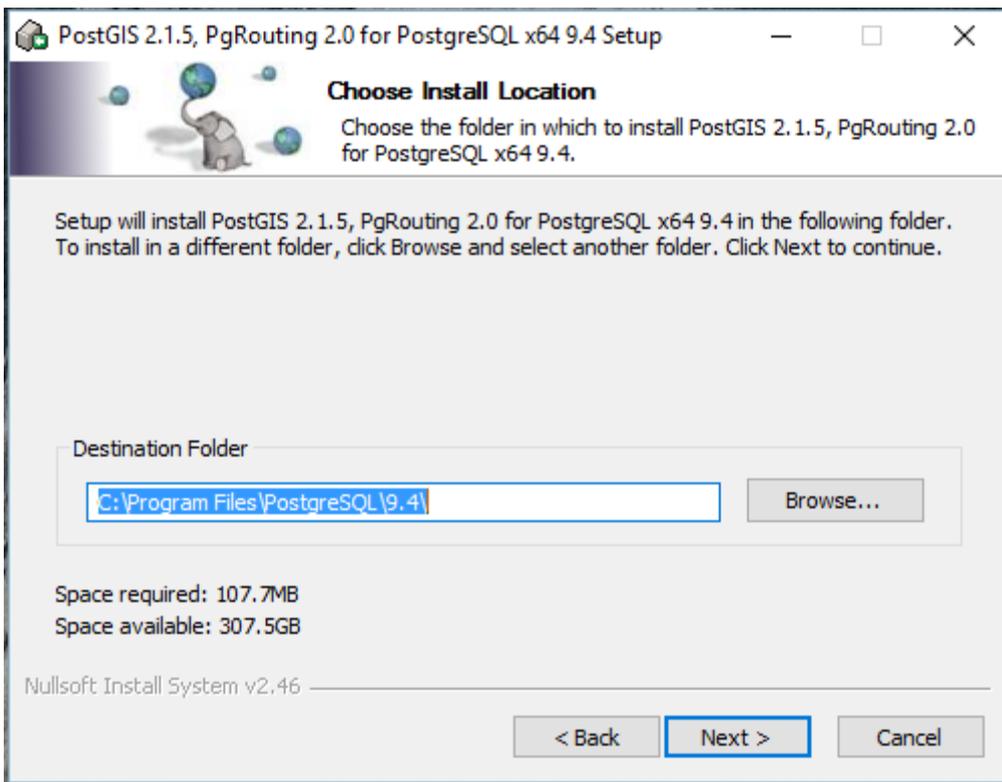
On accepte la licence en cliquant sur « I Agree »



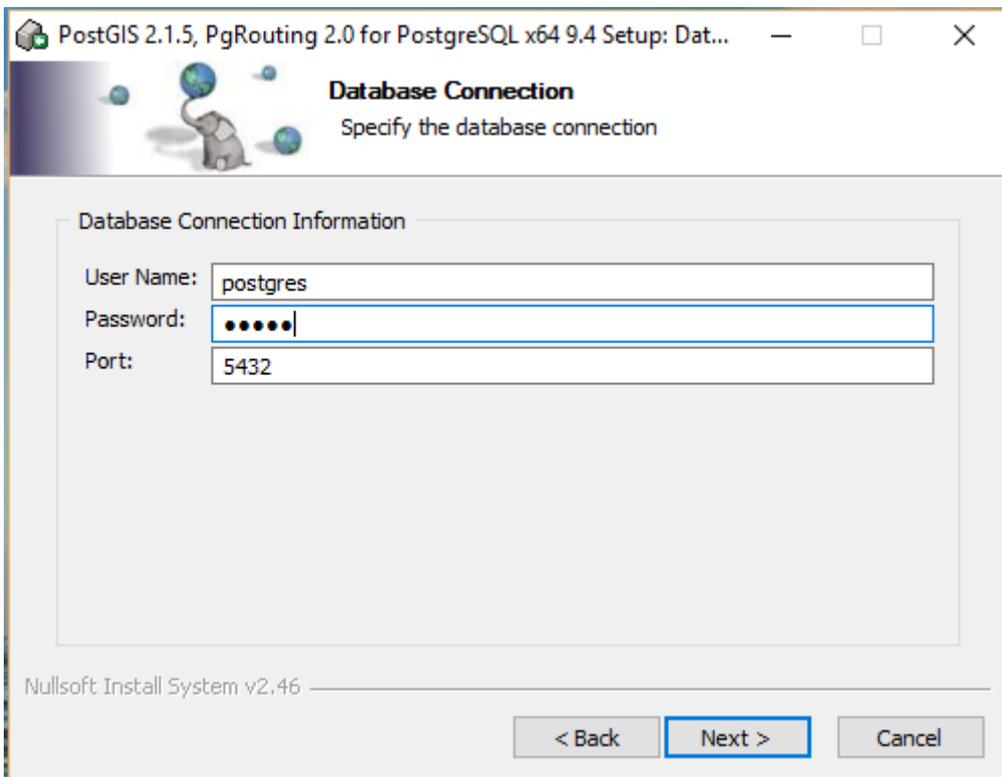
Laisser tous les composants sélectionnés et cliquer sur « Next ».



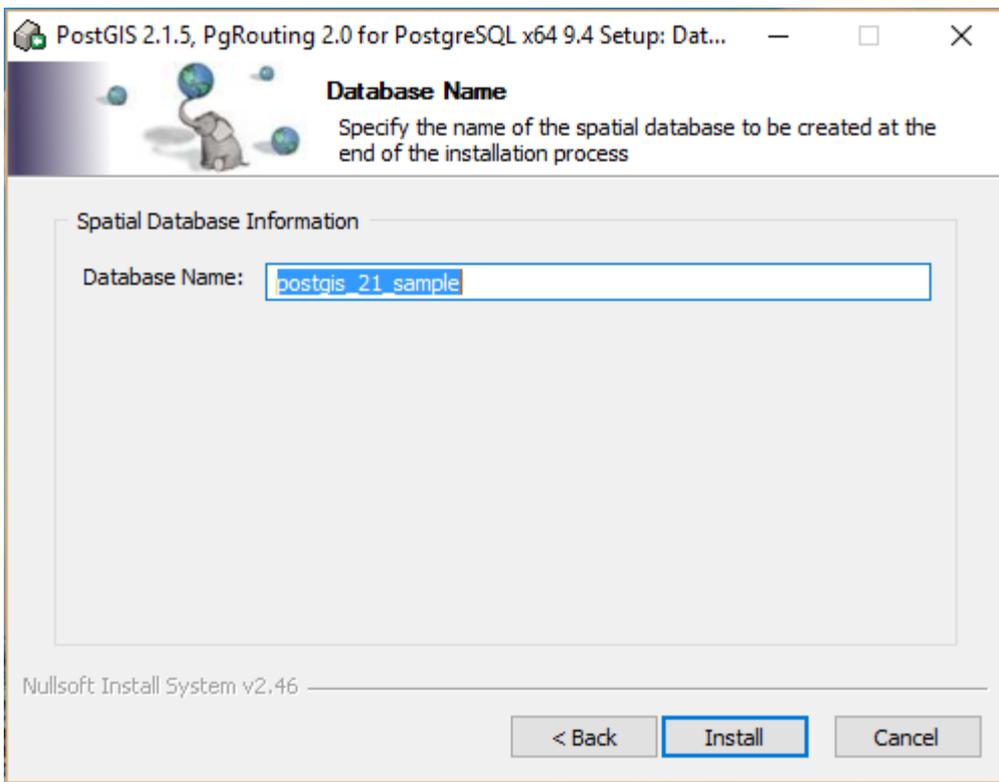
Vérifier que le chemin vers PostgreSQL est correct et cliquer sur « Next ».



Ensuite entrer le mot de passe de l'utilisateur « postgres » et cliquer sur « Next ».



Et enfin, on clique sur « install »



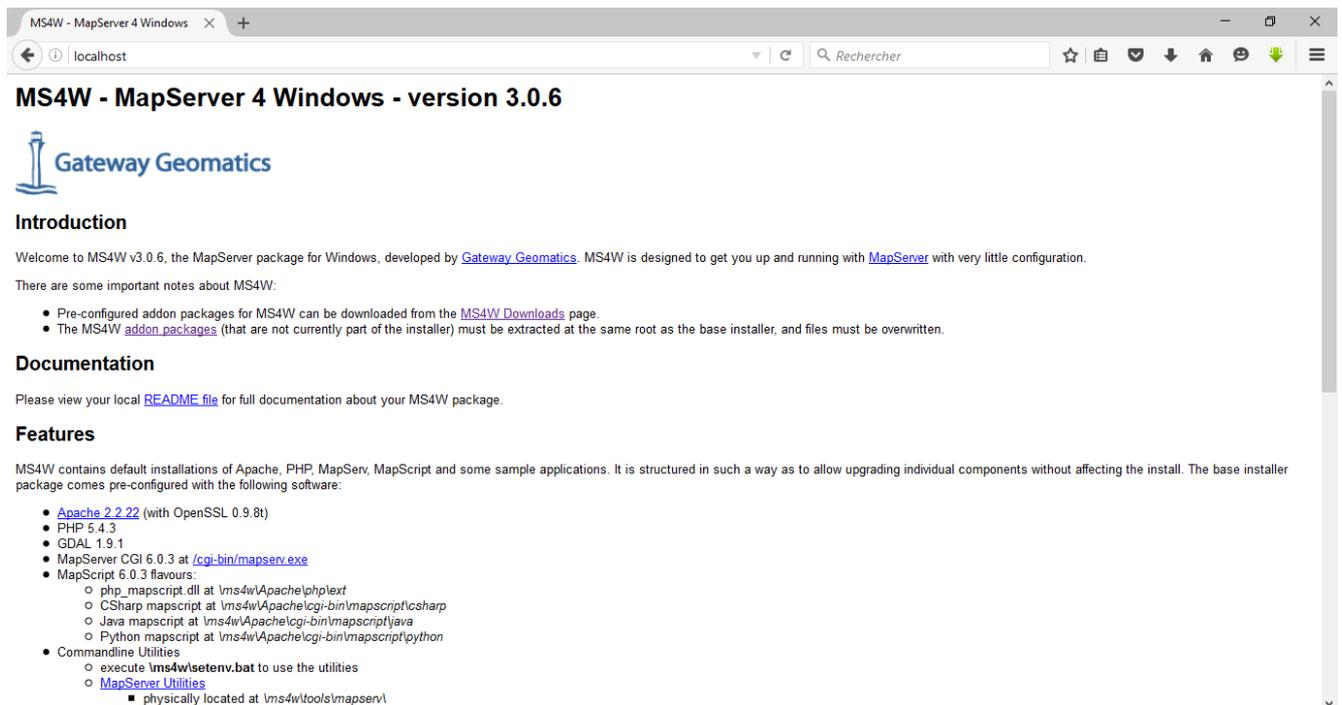
## Annexe 2 : Installation de Mapserver et Pmapper

Après avoir téléchargé le bloc « MS4W » sous forme fichier compressé, il faut l'extraire dans le disque « C : ». Ensuite décompresser « PMAPPER » dans le même répertoire que MS4W en remplaçant certaines composantes.

Lancer « apache-install.bat » en l'exécutant en tant qu'administrateur.

Ensuite double clic sur « setenv.bat ».

A titre de vérification, nous devons lancer un navigateur de notre choix en indiquant l'adresse suivante : <http://localhost/> ou <http://127.0.0.1/>. La capture d'écran suivante sera le résultat :



## Annexe 3 : Description du mapfile

### **Le bloc MAP**

C'est le bloc racine du mapfile, il va contenir les paramètres concernant la globalité de la carte à générer.

Il contient les paramètres suivants : NAME, UNITS, EXTENT, STATUS, FONTSET et SYMBOLSET, SHAPEPATH, IMAGETYPE, SIZE, RESOLUTION, IMAGECOLOR.

#### ***Paramètres généraux :***

**NAME :** c'est le nom de la carte, préfixe ajouté à tous les fichiers images générés par le mapfile, donc à garder de courte taille, et entre guillemets.

**UNITS :** L'unité de la carte, utilisée pour les calculs et le dessin de la barre de l'échelle.

L'information sur l'unité du fond de carte n'est pas toujours apportée par le bloc « PROJECTION », il faut essayer donc de le fournir explicitement à chaque fois.

**EXTENT :** L'extension de la carte, donc coordonnées des extrémités de la carte dessinée dans la carte générée par mapfile.

Ordre : [xmin] [ymin] [xmax] [ymax]

Pour déterminer ces valeurs, il fallait utiliser l'utilitaire « ogrinfo » qui se situe dans le répertoire C:/ms4w/Apache/cgi-bin/.

Avant d'exécuter la commande, les fichiers shapes doivent se trouver dans ce même répertoire.

La figure ci-dessous illustre ce principe :

```

C:\ns4u>cd apache
C:\ns4u\apache>cd cgi-bin
C:\ns4u\apache\cgi-bin>ogrinfo -al -so regions.shp
INFO: Open of 'regions.shp'
      using driver 'ESRI Shapefile' successful.

Layer name: regions
Geometry: Polygon
Feature Count: 22
Extent: (64873.999808, 57715.999623) - (834849.000192, 1566418.000377)
Layer SRS WKI:
PROJCS["laborde",
  GEOGCS["GCS_Tananarive_1925",
    DATUM["Tananarive_1925",
      SPHEROID["International_1924",6378388.0,297.011,
        PRIMEM["Greenwich",0.0],
        UNIT["Degree",0.017453292519943311],
      PROJECTION["Hotine_Oblique_Mercator"],
      PARAMETER["False_Easting",400000.0],
      PARAMETER["False_Northing",0.00000.0],
      PARAMETER["Scale_Factor",0.9995],
      PARAMETER["Azimuth",18.9],
      PARAMETER["Longitude_Of_Center",-46.437229166666],
      PARAMETER["Latitude_Of_Center",-18.9],
      PARAMETER["rectified_grid_angle",18.9],
      UNIT["Meter",1.0],
    REGION: String (30.0)
    ME_FIU: Real (16.0)
    POP93: Real (16.0)
    MCA: String (16.0)
    surface: Real (13.11)
    Province: String (50.0)
  ]

```

**STATUS** : On ou Off, active ou désactive la totalité du mapfile.

**FONTSET et SYMBOLSET** : chemin et nom de fichier contenant les définitions de polices de caractères et de symboles. On peut aussi définir directement un symbole dans le mapfile, mais c'est utile de pouvoir les rassembler dans un seul fichier qui pourra être appelé par différents mapfiles.

**SHAPEPATH** : chemin vers les répertoires contenant les fichiers de données (des couches), ou le chemin de base à partir duquel les paramètres DATA iront les chercher.

**Paramètres de génération de l'image**

**IMAGETYPE** : type de fichier image à générer. Correspond à l'ensemble inclus par défaut dans le bloc output format. Ce paramètre dépend des options avec a été compilé MapServer.

Valeurs : [gif|png|jpeg|bmp|gtiff|swf|user\_defined]

**SIZE** : dimensions (largeur et hauteur) de l'image à générer, en pixels, séparées par un espace. En général, ce sont des valeurs de la largeur et de la hauteur de l'image.

**RESOLUTION** : résolution de l'image en dpi, par défaut c'est 72. N'affecte que les calculs d'échelle.

**IMAGECOLOR** : couleur de fond de l'image de la carte (RGB).

Le bloc MAP va contenir, en plus des paramètres généraux, les blocs de second niveau, qui vont permettre de spécifier le contenu de la carte.

La figure ci-dessous nous montre un exemple concret d'un bloc MAP :

```

16 #
17 # Start of map file
18 #
19 MAP
20 EXTENT 64873.999808 57715.999623 834849.000192 1566418.000377
21
22 UNITS meters
23 #EXTENT -15 30 40 70
24 #UNITS dd
25 SIZE 600 500
26 SHAPEPATH "../../pmapper_demo"
27 SYMBOLSET "../common/symbols/symbols-pmapper.sym"
28 FONTSET "../common/fonts/msfontset.txt"
29 RESOLUTION 96
30 IMAGETYPE png
31 INTERLACE OFF
32 #CONFIG "PROJ_LIB" "C:/proj/nad/"
33 PROJECTION
34 # ETRS-LAEA
35 # "init=epsg:3035"
36 # "+proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000 +ellps=GRS80 +units=m +no_defs no_defs"
37 "init=epsg:29700"
38 "init=epsg:29702"
39 END
40

```

Normal text file      length: 13457    lines: 661      Ln: 91    Col: 17    Sel: 0      UNIX      ANSI as UTF-8    INS

### Le bloc WEB :

Ce bloc va paramétrer le fonctionnement du serveur web autour de la carte générée par MapServer.

**EMPTY [url] :** URL à renvoyer à l'utilisateur si une requête échoue. Si elle n'est pas définie la valeur du paramètre ERROR est utilisée.

**ERROR [url] :** URL à renvoyer à l'utilisateur si une erreur arrive. Les vieux messages moches d'erreur de MapServer apparaîtront si la valeur n'est pas définie.

**FOOTER [nom de fichier] :** Modèle à utiliser après que tout ait été envoyé. En mode requête de plusieurs résultats seulement.

**HEADER [nom du fichier] :** Modèle à utiliser AVANT que tout soit envoyé. En mode requête de plusieurs résultats seulement.

**IMAGEPATH [chemin] :** Chemin vers le répertoire temporaire pour l'écriture des fichiers et images temporaires. L'écriture doit être possible par l'utilisateur du serveur web.

Doit se terminer par un / ou en fonction de la plateforme.

**IMAGEURL [chemin]** : URL de base pour IMAGEPATH. C'est l'URL qu'utilisera le navigateur web vers IMAGEPATH pour obtenir les images

### Le bloc REFERENCE

Trois types de cartes de référence sont gérés. La plus commune serait celle montrant l'étendue d'une carte dans une interface interactive. Il est également possible d'interroger des cartes de référence comme partie d'une requête. Les requêtes ponctuelles généreront une image avec un marqueur (voir plus bas) placé sur le point de requête. Les requêtes basées sur une région montreront l'étendue de la zone d'intérêt. Enfin, les requêtes basées sur les features afficheront la sélection des features utilisées.

**COLOR [r] [g] [b]** : Couleur dans laquelle la boîte de référence est dessinée. Définissez tous les composants a -1 pour avoir une couleur transparente. Rouge par défaut.

**EXTENT [minx][miny][maxx][maxy]** : L'étendue spatial de l'image de base pour la référence.

**IMAGE [nom du fichier]** : Nom du fichier complet de l'image de base pour la référence.

Ce doit être une image GIF

**MARKER [entier|string]** : Définie un symbole (à partir du fichier de symboles) à utiliser lorsque la boîte devient trop petite (voir MINBOXSIZE et MAXBOXSIZE plus bas). Utilise une croix par défaut

**MARKERSIZE [entier]** : Définie la taille du symbole à utiliser au lieu de la boîte (voir MARKER plus haut).

**MINBOXSIZE [entier]** : Si la boîte est plus petite que MINBOXSIZE (utilise la largeur ou la hauteur de la boîte) alors utilise un symbole défini par MARKER et MARKERSIZE.

**MAXBOXSIZE [entier]** : Si la boîte est plus grande que MAXBOXSIZE (utilise la largeur ou la hauteur de la boîte) alors ne dessine rien (souvent l'ensemble de la carte est couvert lorsque vous zoomez en arrière et il est parfaitement évident de l'endroit où vous êtes).

**OUTLINECOLOR [r] [g] [b]** : Couleur à utiliser pour le contour de la boîte de référence.

Définissez une valeur à -1 pour ne pas avoir de contour.

**SIZE [x][y]** : Taille, en pixel, de l'image de base de la référence.

**STATUS [on|off]** : Est-ce que la carte de référence doit être créée ? Off par défaut.

```
#
# Start of Reference map definition
#
REFERENCE
  EXTENT 64873.999808 57715.999623 834849.000192 1566418.000377
  IMAGE "../../../images/reference.png"
  SIZE 199 149
  COLOR -1 -1 -1
  OUTLINECOLOR 255 0 0
END # Reference
```

### Le bloc LEGEND :

La taille de l'image de la légende n'est pas connue avant sa création, soyez donc attentif de ne pas coder en dur la largeur et la hauteur dans la balise <img> du fichier modèle

**IMAGECOLOR [r] [g] [b]** : Couleur d'initialisation de la légende (i.e. l'arrière-plan).

**INTERLACE [on|off]** : [on] par défaut. Ce mot-clé est maintenant déprécié en faveur de l'utilisation de la ligne FORMATOPTION « INTERLACE = ON » dans la déclaration OUTPUTFORMAT.

**LABEL** : Signal le début d'un objet LABEL.

**OUTLINECOLOR [r] [g] [b]** : Couleur à utiliser pour le contour de la boîte des symboles.

**POSITION [ul|uc|ur|ll|lc|lr]** : Où placer une légende incluse dans la carte par défaut.

**KEYSIZE [x][y]** : Taille des boîtes de symboles en pixels. 20 par 10 par défaut.

**KEYSPACING [x][y]** : Espace entre les boîtes de symboles ([y]) et les étiquettes ([x]) en pixels. La valeur par défaut est 5 par 5.

**POSTLABELCACHE [true|false]** : Demande à MapServer de réaliser le rendu de cette légende après que toutes les étiquettes dans le cache aient été dessinées. Utile pour ajouter des bordures ou des éléments similaires. False par défaut.

**STATUS [on|off|embed]** : Est-ce que la légende doit être créée ?

**TEMPLATE [filename]** : Fichier modèle HTML de la légende.

**TRANSPARENT [on|off]** : Est-ce que la couleur d'arrière-plan doit être transparente.

Cette balise est maintenant dépréciée en faveur de la déclaration de la transparence dans les déclarations OUTPUTFORMAT. Off par défaut.

### Le bloc PROJECTION :

L'indication d'un géo référencement pour la carte produite par le mapfile nécessite un bloc PROJECTION « LABORDE » pour Madagascar. Ce bloc contient les informations relatives au référencement, qui peuvent être de deux formes :

```
PROJECTION
'proj=omerc'
'lat_0=-18.9'
'lonc=44.1'
'alpha=18.9'
'k=0.999500000000000001'
'x_0=400000'
'y_0=800000'
'ellps=int1'
'cowgs64=-189,-242,-91,0,0,0,0'
'pm=2.337229166698544
'units=m'
'no_defs'
END
```

OU

```
PROJECTION
-
"init=epsg:29700"
"init=epsg:29702"
END
```

## **Le bloc SCALEBAR**

Mapserver gère les échelles selon une technique assez particulière. En effet, il part du principe que la carte sera une image possédant des dimensions en pixels, qui sera visualisée au moyen d'un écran qui possède une certaine résolution. Par ailleurs, la carte doit être dessinée dans un rectangle d'extension maximale donné par le paramètre EXTENT.

L'échelle finale de la carte doit être donc définie selon ces paramètres. Le paramètre

EXTENT prime sur le paramètre SCALE, car c'est lui qui définit plus précisément ce que doit contenir la carte à dessiner.

Pour dessiner une échelle indiquant une certaine longueur terrain, il faut donc déterminer combien de pixels cette longueur va représenter et l'indiquer dans le paramètre size du bloc.

Le bloc SCALEBAR permet de dessiner des barres d'échelle, dans l'image de la carte ou comme une image distincte. Il possède les paramètres suivants

**POSITION** : Code à deux lettres définissant l'endroit où sera dessinée l'échelle, ce code est le même que celui utilisé par le paramètre POSITION du bloc LEGEND.

**SIZE** : dimensions en pixels (largeur espace hauteur) du rectangle contenant la barre d'échelle. Important car détermine la longueur totale de la barre.

**INTERVALS** : nombre de subdivision à afficher.

**STATUS** : inclusion (EMBED), dans une image à part (ON) ou annulation (OFF).

**STYLE** : apparence de la barre, 0 donnant une barre de rectangle plein de couleurs alternées, 1 une barre fine munie de repères (barbules vers le haut).

**UNITS** : unités pour le calcul de la longueur des intervalles et l'affichage de l'unité de la barre d'échelle. Toutes unités possibles sauf degrés décimaux.

**IMAGECOLOR** : couleur RGB du rectangle qui contient l'échelle.

**BACKGROUNDCOLOR** : couleur RGB de la barre d'échelle et de ses libellés.

**COLOR** : Couleur alternative à BACKGROUNDCOLOR si barre de type 0 et plusieurs intervalles spécifiés.

**OUTLINECOLOR** : Couleur RGB de la réserve autour de la barre d'échelle (mais pas autour des libellés).

**TRANSPARENT** : Valeur booléenne (ON/OFF) qui précise si le rectangle contenant l'échelle est transparent

```
#
# Start of ScaleBar definition
#
SCALEBAR
  STATUS embed
  TRANSPARENT off
  INTERVALS 4
  SIZE 200 3
  UNITS kilometers
  COLOR 250 250 250
  OUTLINECOLOR 0 0 0
  BACKGROUNDCOLOR 100 100 100
  STYLE 0
  POSTLABELCACHE true
  LABEL
    COLOR 0 0 90
    #OUTLINECOLOR 200 200 200
    SIZE small
  END # Label
END # SCALEBAR
```

### Le bloc OUTPUTFORMAT :

Ce bloc permet de définir précisément le format d'image du fichier qui sera généré par MapServer. Le paramètre général **IMAGETYPE CORRESPOND** en fait à des blocs OUTPUTFORMAT prédéfinis dans MapServer, par défaut. On peut ainsi mieux préciser certains paramètres de la sortie image, par exemple le taux de compression du format de l'image.

```
OUTPUTFORMAT
# 8 bit PNG
NAME 'agg_png8'
DRIVER AGG/PNG
IMAGEMODE RGB
FORMATOPTION "QUANTIZE_FORCE=ON"
FORMATOPTION "QUANTIZE_DITHER=OFF"
FORMATOPTION "QUANTIZE_COLORS=256"
END
```

```
OUTPUTFORMAT
NAME GTiff
DRIVER "GDAL/GTiff"
MIMETYPE "image/tiff"
IMAGEMODE RGB
#FORMATOPTION "TFW=YES"
#FORMATOPTION "COMPRESS=PACKBITS"
EXTENSION "tif"
END
```

### Le bloc LAYER :

Ce bloc va définir les propriétés de création et d'affichage d'une couche de données SIG par mapserver. Les blocs LAYER sont dessinés dans l'ordre du mapfile, c'est-à-dire que le premier bloc du mapfile est dessiné en premier, les suivants viendront par-dessus sur la carte.

L'ordre du mapfile est donc l'ordre inverse de la superposition verticale des sources.

**NAME** : nom de la couche, utilisé comme identifiant dans l'interface web. Il doit être unique dans le mapfile et d'une longueur maximale de 20 caractères. Entre guillemets de préférence (ou apostrophes).

**GROUP [name]** : Nom d'un groupe auquel la couche appartient. Le nom de groupe peut ensuite être référencé comme un nom normal de couche dans les fichiers de mise en forme

("Template"), ce qui permet de faire certaines opérations comme afficher ou cacher un groupe de plusieurs couches en une seule opération.

**METADATA** : Ce mot-clé permet d'enregistrer des données arbitraires comme un couple de valeurs. Ceci est utilisé avec OGC WMS pour définir des éléments comme le titre de la couche. Il peut aussi permettre plus de souplesse lors de la création des mises en forme

("Template"), puisque tout ce que vous mettrez ici sera accessible via les balises du Template.

**STATUS [on|off|défaut]** : statut (visibilité) du layer. Doit prendre la valeur « default » pour que le layer soit visible lorsque l'on utilise mapserver en mode map (le statusOn ne suffit pas à le rendre visible, il faut que le layer soit expressément requis).

Définit le statut actuel de la couche. Fréquemment modifié par MapServer lui-même. Default active la couche ("on") en permanence. En mode CGI, les couches avec STATUS DEFAULT ne peuvent pas être désactivées par les mécanismes classiques. Il est recommandé de définir des couches en STATUS DEFAULT lorsque l'on souhaite déboguer un problème puis de les redéfinir à ON/OFF en utilisation normale.

Pour le WMS, les couches dans le mapfile du serveur dont le paramètre STATUS vaut DEFAULT sont toujours envoyées au client.

**TYPE [point|line|polygon|circle|annotation|raster|query|chart]** :

Définie comme les données doivent être dessinées. Ne doit pas être nécessairement le même que celui du type du shapefile. Par exemple, un shapefile de type polygone peut être dessiné comme une couche point, mais un shapefile de type point ne peut pas être dessiné comme une couche polygone. Règle de bon sens. Annotation signifie qu'un point étiquette sera calculé pour les features, mais la feature en elle-même ne sera pas dessinée bien qu'un marqueur peut être optionnellement dessiné. Cela permet l'étiquetage avancé comme numéroter les autoroutes. Les points sont étiquetés de cette manière. Les polygones sont d'abord étiquetés en utilisant un centroïde, et s'il ne tombe pas dans le polygone une approche de "scan de ligne" est utilisée pour garantir que l'étiquette tombe dans la feature.

Les lignes sont étiquetées au milieu de l'arc le plus long dans la portion visible de la ligne.

query signifie seulement que la couche peut être interrogée mais pas dessinée.

Dans le but de différencier les POLYGON et les POLYLINE (qui n'existent pas en tant que type), utilisez ou omettez simplement respectivement le mot-clé COLOR lors de la classification. Si vous l'utilisez, c'est un polygone avec une couleur de remplissage, sinon ce sera une polyligne avec seulement un OUTLINECOLOR.

Un cercle doit être défini par une limite rectangulaire minimale. Pour cela, deux points définissent le plus petit carré qui peut y être contenu. Ces deux points sont les deux coins opposés de cette boîte.

**MINSCALE [double] – déprécié** :

Depuis MapServer 5.0, le bon paramètre à utiliser est MINSCALEDENOM. Le paramètre obsolète MINSCALE était l'échelle minimale à laquelle la couche était dessinée. L'échelle est donnée comme dénominateur de la fraction de l'échelle actuelle. Par exemple pour une carte à l'échelle de 1 :24000, il s'agit de 24000.

**MAXSCALE [double] – déprécié** :

Depuis MapServer 5.0, le bon paramètre à utiliser est MAXSCALEDENOM. Le paramètre obsolète MAXSCALE était l'échelle maximale à laquelle la couche était dessinée.

L'échelle est donnée comme dénominateur de la fraction de l'échelle actuelle. Par exemple pour une carte à l'échelle de 1 :24000, il s'agit de 24000.

### **SYMBOLSCALE [double] - déprécié**

Depuis MapServer 5.0 le paramètre correct à utiliser est SYMBOLSCALEDENOM. Le paramètre SYMBOLSCALE déprécié est l'échelle à laquelle les symboles et/ou les textes apparaissent à leur taille normale. Cela permet une échelle dynamique des objets en fonction de l'échelle de la carte. S'il n'est pas défini alors cette couche apparaîtra toujours à la même taille. La mise à l'échelle a lieu seulement entre les limites comprises entre MINSIZE et

MAXSIZE comme décrit plus haut. L'échelle est donnée comme le dénominateur de la fraction de l'échelle, par exemple, pour une carte à l'échelle de 1 :24 000 utilisez 24000.

Implémenté dans MapServer 5.0, pour remplacer le paramètre déprécié SYMBOLSCALE.

### **OPACITY [entier|alpha] :**

Définit le niveau d'opacité (autrement dit l'incapacité de voir à travers la couche) de tous les pixels classés pour une couche donnée. La valeur peut être soit un entier compris entre 0 et

100 soit le mot-clé nommé "ALPHA". Une valeur de 100 est opaque et une valeur de 0 est complètement transparente. Implémenté dans MapServer 5.0 pour remplacer le paramètre obsolète TRANSPARENCY.

Le mot-clé "ALPHA" indique au moteur de rendu de MapServer de prendre en compte la transparence indexée ou alpha des symboles pixmap utilisés pour styler une couche. C'est uniquement nécessaire en cas de formats de sortie RVB et ne devrait être utilisé que lorsque cela est nécessaire car il est coûteux de rendre transparents des symboles pixmap sur une image de carte RGB.

### **OFFSITE [r] [g] [b] :**

Sélectionne l'index de couleur utilisé pour la transparence des couches raster.

### **POSTLABELCACHE [true|false] :**

Indique à MapServer de dessiner cette couche après que toutes les étiquettes en cache auront été dessinées. Très utile pour ajouter une touche finale et d'autres éléments similaires.

Désactivé par défaut.

### **CLASSITEM [attribut] :**

Nom de l'élément dans la table attributaire à utiliser dans les correspondances de classe.

### **LABELITEM [attribut] :**

Nom de l'élément dans la table attributaire à utiliser comme classe d'annotation (par exemple pour l'étiquetage).

## **TEMPLATE [fichier|url] :**

Utilisé comme une alternative globale au TEMPLATE CLASS.

### ***Paramètres de données :***

#### **Shapefiles**

Les shapefiles sont les sources de données la plus courante pour mapserver, qui au départ était construit uniquement autour de ce format (issu de la bibliothèque shapelib). On indique que la source de données du layer est un shapefile avec le paramètre DATA, qui prend comme valeur le chemin vers le shapefile, relatif au chemin SHAPEPATH. Le paramètre FILTER permet d'utiliser les expressions régulières pour filtrer les objets de la source de données au niveau du bloc LAYER, c'est-à-dire avant le traitement des blocs CLASS qui peuvent eux aussi comporte une sélection.

L'utilisation d'une source de données shapefile offre de plus la possibilité de lier (jointure) une table dbf de données attributaires, en utilisant un bloc secondaire JOIN. Ces données deviennent ainsi interrogeables par mapserver en mode Query.

#### **Couches vectorielles accédées avec OGR**

La bibliothèque de fonctions OGR permet à mapserver de lire un plus grand nombre de formats de SIG vectoriels

Pour indiquer une source de données OGR, il faut utiliser la syntaxe suivante :

CONNECTIONTYPE OGR

CONNECTION [nom de la source]

Ce dernier paramètre dépend du type de source, généralement il s'agit du chemin relatif au

SHAPEPATH et du nom de fichier ou du nom de répertoire. Certains types de données vectorielles sont organisés en couches multiples par fichier, il faut donc choisir la couche à utiliser avec le paramètre DATA [numero/nom de la couche]. Mapserver peut aussi récupérer en partie les éventuels styles d'affichage présents dans les couches, avec le paramètre

STYLEITEM AUTO. Le paramètre FILTER est utilisable aussi avec ce type de données, comme pour les shapefiles.

#### ***Sources serveurs de données***

Mapserver peut se connecter à des serveurs SIG directement, et ainsi assembler des couches de diverses origines de façon transparente. Cela permet de construire une organisation des données dispersées sur des serveurs spécialisés. Les paramètres de connexion sont fournis dans une chaîne de caractères passée en valeur au paramètre CONNECTION. Le paramètre FILTER peut contenir une expression de requête SQL c'est-à dire le texte venant après un élément SQL WHERE.

#### **Syntaxe de connexion entre Mapserver et les serveurs de données Postgis :**

CONNECTIONTYPE postgis

CONNECTION « user=nom\_utilisateur dbname=nom\_bdd host=serveur »

DATA « colonne\_géométrique FROM table\_ou\_requête\_SQL »

### **Oracle Spatial :**

CONNECTIONTYPE oraclespatial

CONNECTION « nom\_utilisateur/mot\_de\_passe@bdd »

DATA « colonne\_géométrique FROM table\_ou\_requête\_SQL »

### **Couches raster**

Les données raster géo référencées sont gérées avec le paramètre TYPE raster, le paramètre

DATA devant pointer vers le fichier (et son éventuel chemin relatif à SHAPEPATH).

Mapserver possède quelques fonctions de traitement des rasters, accessibles avec le paramètre

PROCESSING. GDAL est capable de réaliser des traitements de rééchantillonnage, mise à l'échelle, de tramage et de sélection de bande.

Les sources de données rasters composés de nombreux fichiers juxtaposés sont gérables en une seule fois par Mapserver, sous la forme d'un tuilage. Ce tuilage consiste en un fichier

shape qui contient des objets polygone représentant l'extension de chaque tuile raster, et ayant comme attribut le nom du fichier. Ce type de tuilage peut être produit en utilisant l'utilitaire gdalindex distribué avec GDAL. Cela permet une gestion beaucoup plus rapide des grandes couvertures raster, notamment si les bitmaps sont eux aussi optimisés pour être lus à différentes résolutions (index pyramidal dans les tifs par exemple).

### **Le bloc CLASS**

Ce bloc permet de définir des classes thématiques dans la couche, qui vont pouvoir être affichées différemment sur la carte globale. Les blocs CLASS sont traités dans l'ordre du fichier map, selon l'ordre de classement vertical. Ce bloc peut contenir les paramètres suivants :

NAME [string]

Nom à utiliser dans les légendes pour cette CLASS. Si ce n'est pas défini, la CLASS n'apparaîtra pas dans une légende.

### **EXPRESSION [string] :**

Quatre types d'expressions sont maintenant supportés pour définir l'appartenance à une classe. Les comparaisons de chaînes, les expressions régulières, les expressions logiques simples et les fonctions de chaînes. Si aucune expression n'est donnée, alors toutes les entités sont considérées comme appartenant à cette classe.

Les comparaisons de chaînes sont sensibles à la casse et sont les plus rapides à analyser. Il n'y a pas de caractère délimiteur à utiliser cependant la chaîne doit être entre quotes si elle contient des caractères spéciaux (Afin de prendre de bonnes habitudes, il est recommandé de mettre entre quotes toutes vos chaînes).

Les expressions régulières fonctionnent de la même façon que dans les précédentes versions de MapServer. Cependant vous devez dorénavant délimiter toute expression régulière en utilisant /regex/. Aucune quote ne doit être utilisée.

Les expressions logiques vous permettent de construire des tests assez complexes basés sur un ou plusieurs attributs et sont donc seulement utilisables avec les fichiers Shape. Les expressions logiques sont délimitées par des parenthèses "(expression)".

Les noms des attributs sont délimités par des crochets "\*ATTRIBUT+". Ces noms sont sensibles à la casse et doivent correspondre aux objets dans le fichier Shape

### **COLOR [r] [g] [b] :**

Couleur à utiliser pour dessiner les entités.

### **OUTLINECOLOR [r] [g] [b] :**

Couleur à utiliser pour les polygones et certains symboles de marquage. Les symboles de ligne ne gèrent pas ce paramètre.

### **SYMBOL [integer|string|filename] :**

Le nom du symbole ou le numéro à utiliser pour toutes les entités si les tables d'attributs ne sont pas utilisées. Le numéro est l'index du symbole dans le fichier de symboles, celui-ci commence à 1, le 5ième symbole dans le fichier est donc le symbole numéro 5. Vous pouvez aussi donner des noms à vos symboles en utilisant le mot-clé NAME dans le fichier de définition des symboles puis l'utiliser pour y faire référence. 0 est la valeur par défaut ce qui produit un simple pixel, une ligne simple fine ou un polygone plein selon le type de la couche.

Vous pouvez aussi spécifier un fichier GIF ou PNG. Le chemin est relatif à l'emplacement du mapfile.

### **SIZE [integer] :**

Hauteur, en pixels, du symbole/motif à utiliser. Seulement utile avec les symboles reéchantillonnables.

Pour les symboles de type vecteur (et ellipse), la taille par défaut est basée sur la gamme des valeurs de Y dans les POINTS définissant le symbole. Pour les rasters, le défaut est la taille verticale de l'image. La taille par défaut est de 1 pour les symboles TTF

**MINSIZE [integer] :**

Taille minimale en pixels pour dessiner un symbole. Le défaut est de 0.

**MAXSIZE [integer] :**

Taille maximale en pixels pour dessiner un symbole. Le défaut est de 50.

**TEXT [string] :**

Texte fixe pour étiqueter les entités de cette classe. Ceci écrase les valeurs obtenues par

LABELITEM. La chaîne de caractères peut être donnée comme une expression délimitée par des parenthèses. Ceci vous permet de concaténer plusieurs attributs en une seule étiquette. Par exemple : ([FIRSTNAME], [LASTNAME])

**TEMPLATE [filename] :**

Fichier template ou URL à utiliser pour présenter les résultats des requêtes à l'utilisateur.

Voyez la référence *Templating* pour plus d'information.

**DEBUG [on|off] :**

Active le débogage de l'objet classe. Une sortie verbeuse est générée et envoyée au canal de sortie standard des erreurs (STDERR) ou vers le fichier de trace ("log") de MapServer si celui-ci est précisé dans le paramètre LOG de l'objet *WEB*.

Le bloc CLASS peut contenir des blocs secondaires LABEL et STYLE.

**Le bloc LABEL :**

Ce bloc permet de configurer l'étiquetage des éléments de la classe, selon un champ attributaire indiqué par le paramètre LABELITEM du bloc LAYER qui contient le bloc LABEL.

**Paramètres de base :**

**TYPE [bitmap|truetype] :**

Type de police de caractères à utiliser. Généralement les polices bitmap sont plus rapides à dessiner que les polices TrueType. Cependant les polices TrueType sont redimensionnables et disponibles sous de nombreuses formes. Assurez-vous que le paramètre FONT est activé si vous choisissez "TrueType".

**COLOR [r] [g] [b] | [attribute] :**

Couleur à utiliser pour dessiner le texte.

[ATTRIBUTE] a été implémenté dans la version 5.0 pour spécifier le nom de l'attribut dans la table attributaire à utiliser pour les valeurs de couleur. Les crochets [] sont nécessaires

**SIZE [integer][[tiny|small|medium|large|giant]][[attribute]]:**

Taille du texte. Utiliser “integer” pour donner la taille en pixels de votre police de caractères

TrueType ou utilisez l’un des 5 mots-clés listés pour les polices de caractères bitmap.

[*Attribute*] a été implémenté dans la version 5.0 pour spécifier le nom de l’attribut dans la table attributaire à utiliser pour les valeurs de taille. Les crochets [] sont nécessaires.

**MINSIZE [integer] :**

Taille minimale de la police de caractères à utiliser pour redimensionner le texte (en pixels). 4 par défaut.

**MAXSIZE [integer] :**

Taille maximale de la police de caractères à utiliser pour redimensionner le texte (en pixels).

256 par défaut.

**MINFEATURESIZE [integer|auto] :**

Taille minimale d’une entité devant être étiquetée, en pixels. Pour les données linéaires, la longueur totale de la ligne affichée est utilisée, pour les entités polygonales la taille du plus petit côté du rectangle englobant est utilisée. Le mot-clé “Auto” indique à MapServer d’étiqueter uniquement les entités plus grandes que leur propre étiquette.

Disponible seulement pour les étiquettes en cache.

***Paramètres d’effets d’affichage du texte (la présence du paramètre active la fonction) :***

**ANTIALIAS [true|false] :**

Le texte doit-il être antialiasé ? Notez que ceci nécessite un plus grand nombre de couleurs, diminue les performances de rendu et génère des images plus lourdes.

**OUTLINECOLOR [r] [g] [b] | [attribute] :**

Couleur utilisée pour dessiner une ligne d’un pixel autour du texte.

[*ATTRIBUTE*] a été implémenté dans la version 5.0 pour spécifier le nom de l’attribut dans la table attributaire à utiliser pour les valeurs de couleur. Les crochets [ ] sont nécessaires.

**OUTLINEWIDTH [integer] :**

Largeur du contour si OUTLINECOLOR a été définie. 1 par défaut. Pour l’instant sur le moteur de rendu AGG gère des valeurs supérieures à 1, et rend un contour avec un effet

“d’halo” : valeurs recommandées : 3 ou 5.

**SHADOWCOLOR [r] [g] [b] :**

Couleur de l’ombre.

### **SHADOWSIZE [x][y] :**

Décalage de l'ombre en pixels.

### **BACKGROUNDCOLOR [r] [g] [b] :**

Couleur à utiliser pour dessiner un rectangle d'arrière-plan (panneau d'étiquette ("billboard")). Désactivé par défaut.

### **Paramètres de positionnement :**

#### **POSITION [ul|uc|ur|cl|cc|cr|ll|lc|lr|auto] :**

Position de l'étiquette relative au point d'insertion (couches seulement). La première lettre est la position en "Y", la seconde lettre est la position en "X". "Auto" indique à MapServer de calculer une position d'étiquette qui n'interférera pas avec d'autres étiquettes. Avec des points, MapServer sélectionne l'une des 8 positions extérieures (c'est à dire en excluant cc).

Avec les polygones, MapServer sélectionne cc (ajouté dans MapServer 5.4), uc, lc, cl et cr comme positions possible. Avec des lignes, il utilise seulement lc ou uc, jusqu'à ce qu'il trouve une position qui n'entraîne pas de chevauchement (collision) avec des étiquettes déjà dessinées. Si toutes les positions entraînent un conflit, alors l'étiquette n'est pas dessinée (sauf si le paramètre d'étiquette FORCE est activé). "Auto" est seulement disponible avec les étiquettes en cache.

#### **ANGLE [double|auto|follow|attribute] :**

Angle, données en degrés, pour dessiner l'étiquette.

AUTO permet à MapServer de calculer l'angle. Valide uniquement pour les couches de type ligne (LINE).

FOLLOW a été implémenté dans la version 4.10 et indique à MapServer de calculer une étiquette courbe appropriée aux entités linéaires (cf. la *MS RFC 11 : Support for Curved*

*Labels* pour les détails).

[ATTRIBUTE] a été implémenté dans la version 5.0 pour spécifier le nom de l'attribut dans la table attributaire à utiliser pour les valeurs angulaires. Les crochets [] sont nécessaires

#### **OFFSET [x][y] :**

Valeurs de décalage pour les étiquettes, relatives au coin inférieur gauche de l'étiquette ou du point d'insertion de l'étiquette, en pixels. Dans le cas d'un texte tourné (rotation), cette commande spécifie les valeurs comme si toutes les étiquettes étaient horizontales et toute rotation sera compensée.

#### **MINDISTANCE [integer] :**

Distance minimale entre des étiquettes en double, en pixels.

**BUFFER [integer] :**

Remplissage, en pixels, autour des étiquettes. Utile pour maintenir un espace autour du texte pour en améliorer sa lecture. Disponible seulement pour les étiquettes "cachables".0 par défaut.

**FORCE [true|false] :**

Force l'affichage des étiquettes d'une classe particulière, sans s'occuper des superpositions (collisions). Disponible seulement pour les étiquettes en cache. Désactivé par défaut.

**PARTIALS [true|false] :**

Le texte peut-il être caché s'il n'apparaît qu'en partie sur la carte ? Activé par défaut.

**WRAP [character] :**

Caractère qui représente une rupture de ligne dans le texte de l'étiquette et qui entraîne la création d'une étiquette multi-ligne. Interagit avec MAXLENGTH pour le retour à la ligne conditionnel après un nombre de caractères donnés

```
LABEL
  POSITION Auto
  COLOR 0 0 150
  BUFFER 2
  TYPE truetype
  ENCODING "UTF-8"
  FONT FreeSans
  SIZE 8
  MAXSIZE 9
  #SIZE small
END #Label
```

**Utilisation du Template :**

Dans le cadre d'une utilisation sur un serveur web, les auteurs de MapServer ont ajouté une fonctionnalité au logiciel pour lui permettre de générer non seulement une image de carte, mais un document HTML complet pouvant comprendre les divers éléments comme la carte, l'échelle, la légende, les titres et les mentions. Ce fonctionnement utilise un fichier modèle

HTML, dans lequel MapServer va modifier des valeurs pour les remplacer par les éléments qu'il va produire. Ce fichier modèle HTML est ce qu'on appelle un „Template“ dans le vocabulaire de mapserver.

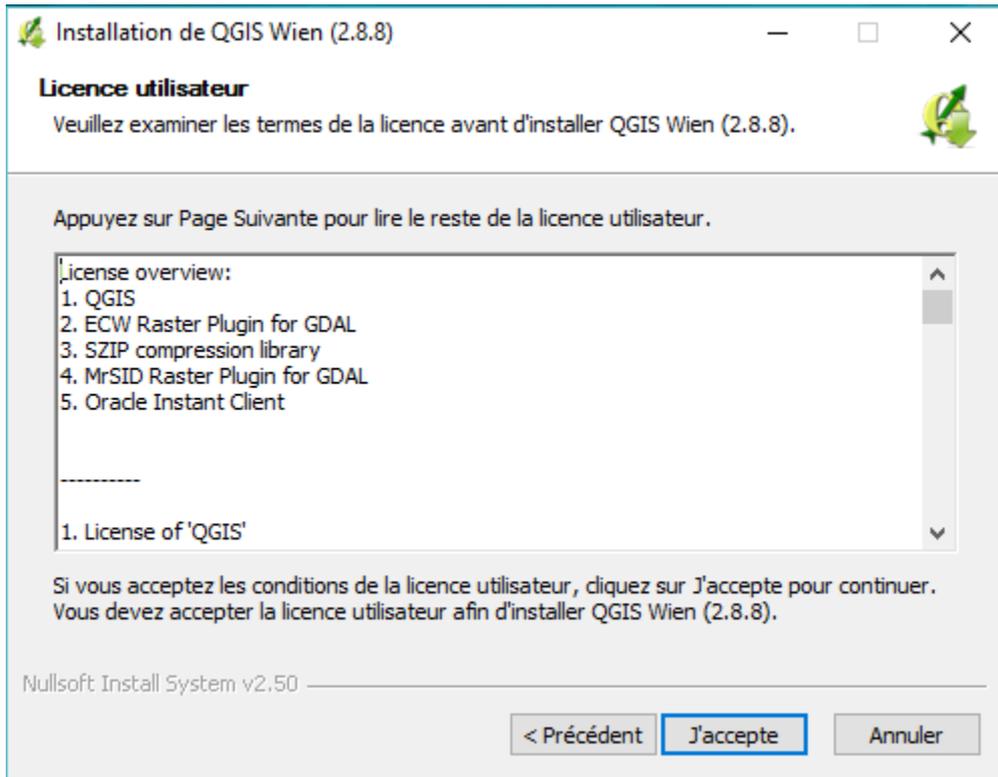
Le fonctionnement des templates est très simple dans le mapfile on spécifie le chemin vers un fichier template HTML. Ce dernier contient le code HTML de la page dans laquelle vont se trouver les éléments produits par mapserver, ainsi que certaines balises spéciales, qui seront remplacées par le code HTML adéquat par mapserver. La balise la plus importante est la balise [img], qui sera remplacée par mapserver par le chemin vers le fichier image de la carte qu'il génère.

## Annexe 4 : Installation de QGIS 2.8

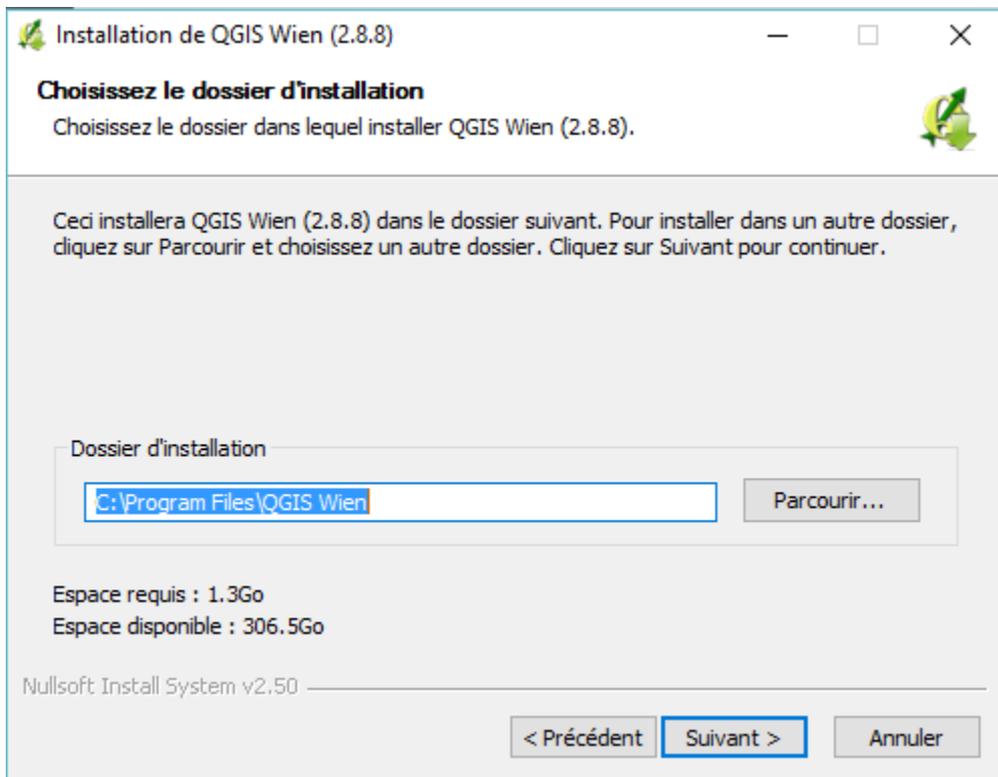
Après téléchargement de QGIS 1.8.0, double clic sur « Setup.exe ». Une fenêtre de bienvenue apparaît, puis clique sur suivant



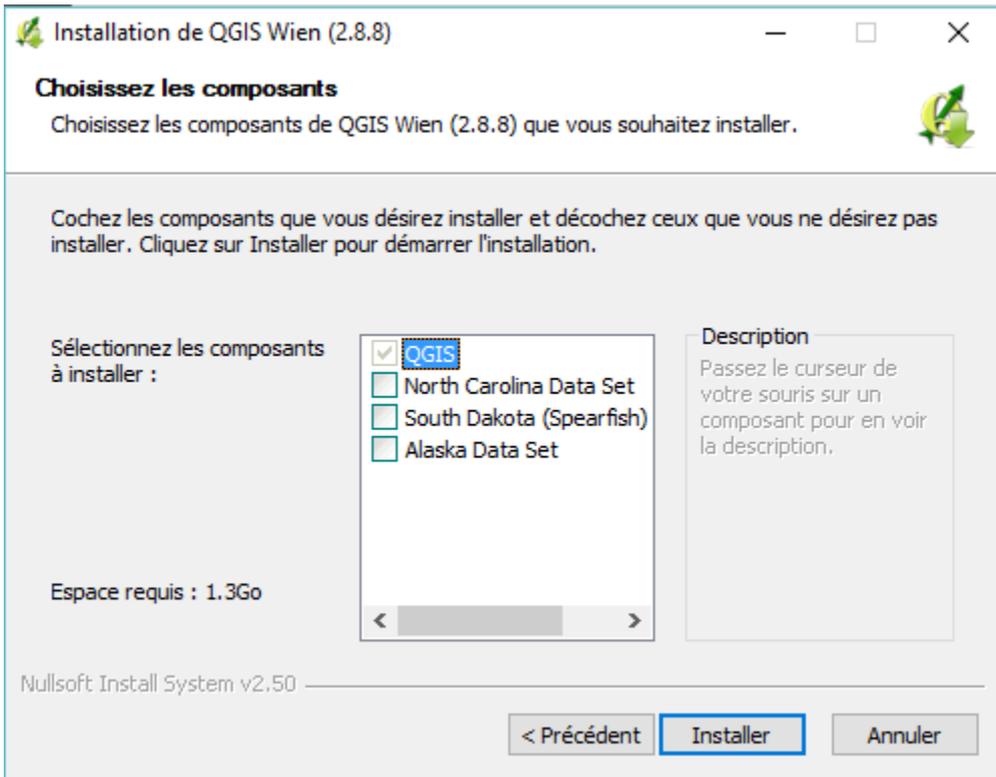
Ensuite en clic sur « j'accepte »



Voici le répertoire d'installation par défaut, après cliquer sur « suivant »



Puisque nous voulons installer simplement QGIS, donc accepter la composante à installer par défaut.



Enfin, si l'installation est finie, on clique sur « Fermer »

