

## Part I

# Présentation générale du problème

## 1 Projet

### 1.1 Finalité

L'objectif de ce projet est de créer une base de données géographiques. Cette base est destinée à constituer un rassemblement volumineux de données géographiques sur le territoire. Les données sont récupérées de sources diverses puis traitées automatiquement. Enfin, elles sont importées dans la base.

À des fins d'études recroisant ces données, cette base est vouée à être une source de données commune à de multiples utilisateurs. Dans cette optique, cette base sera munie d'un ensemble de requêtes basiques, c'est-à-dire de filtres larges permettant une utilisation facilitée de ces données par des logiciels de traitements, en aval.

### 1.2 Problématique

À l'IFSTTAR, certaines recherches requièrent la manipulation de données géographiques : statistiques, recroisements, simulations. Actuellement le regroupement, stockage, filtrage, et partage de ces données est réalisé manuellement. Par conséquent, les mêmes manipulations sont souvent répétées, ce qui constitue une perte d'efforts et de temps.

### 1.3 État des lieux et du besoin

Le travail à effectuer vise à créer un serveur de données centralisé en important automatiquement des données, en les pré-traitant, puis en les récupérant dans la base.

L'utilisateur doit pouvoir définir l'ensemble des paramètres d'exécution avant le lancement. Ainsi, sans modifier le code source, il doit être en mesure de spécifier les éléments suivants ci-dessous.

**comportement :** réaction du programme à une situation donnée (exemple : niveau de verbosité)

**structure de la base :** tables, attributs, cls, relations, contraintes

**donnes :** source des donnes et traitements attendus (exemple : spcifier la table qui va accueillir un ensemble de donnes)

L'ensemble de ces spcifications s'effectueront travers des fichiers de configuration, de format JSON. Toute interaction entre le logiciel et l'utilisateur s'effectuera par ces fichiers. Ainsi, l'utilisateur n'aura pas besoin d'altrer le code source pour contrler le comportement du logiciel.

De plus, seuls les paramtres absolument ncessaires seront prciser, comme par exemple la structure de la base si l'utilisateur souhaite la crer. L'utilisateur aura bien entendu la libert de redfinir les autres paramtres au cas o leurs valeurs par dfaut ne lui conviennent pas.

## 1.4 Solution apportee

Pour rpondre au besoin nonc ci-dessus, une base de donnes centralise ainsi que des procdures d'importation, de nettoyage, et de traitement de donnes sont cres. L'ensemble de ces solutions permet notamment :

- un stockage massif performant de donnes
- l'limination des tches de pr-traitement : rcupration, nettoyage, uniformisation, recherche de liens (cls & relations)
- la mutualisation des donnes, rendues accessibles de multiples utilisateurs
- la mise disposition de requetes basiques de slection avec application de filtres (attributaires, gographiques)

## 1.5 Planification

Diagramme de Gantt :

[height=25em]gantt

## 2 Contexte

### 2.1 tudes dj effectues

Marc Nol a d'ores-et-dj effectu un travail prliminaire sur les donnes et tabli les liens existants. En effet, en parcourant et effectuant des traitements sur ces donnes il a pu conevoir le schma de la base de donnes.

C'est ce schma qu'il revient ici d'implmenter, en discutant des adaptations mineures ventuelles. Ce schma se veut proche des donnes d'entre. De cette manire, l'intgration des donnes est facilite.

### 2.2 Nature des prestations demandes

En premier lieu, le dveloppement de deux scripts shell (**bash**). Ces scripts, non interactifs, vont venir prparer l'excution ultrieure du code **Python**. En effet, ils se chargent de d'installer et configurer automatiquement l'ensemble des composants de l'environnement. Le premier se charge d'installer et de configurer la base **PostgreSQL**. Le second se charge de prparer l'excution du code **Python** en installant les dpendances (les packages) auxquels il fait appel.

En second lieu, le dveloppement d'un script **Python** charg de manipuler les donnes. Ce dernier doit effectuer un ensemble de traitements tels que preiss par l'utilisateur.

### 2.3 Parties concernees par le droulement du projet et ses rsultats

En ce qui concerne l'IUT informatique de Nantes, Monsieur Log Jezequel, est charg en tant qu'enseignant rfrent de suivre et valuer ce stage. Ce suivi est ralis via un message lectronique envoy en chaque fin de semaine, ainsi qu'une visite sur le lieu de stage. De mme, plusieurs rendus sont fournir tel qu'un rsum du sujet, ce cahier des charges, ainsi que le rapport de stage.

En ce qui concerne la structure d'accueil, l'IFSTTAR, Pascal Gastineau et Pierre Hankach jouent le rle de tuteurs. Ils seront galement les commanditaires et les clients du dveloppement effectu.

Enfin, une soutenance ponctue le stage. Le jury sera alors compos de Pierre Hankach, Log Jezequel, ainsi qu'un second professeur de l'IUT.

## Part II

# Dtail technique du besoin

### 3 tapes prliminaires : scripts shell

#### 3.1 Installation et configuration de PostgreSQL

En premier lieu, le rle de ce script est d'installer PostgreSQL et l'extension PostGIS.

En second lieu, ce script a pour rle de configurer PostgreSQL. Cela est ralis via l'dition automatise de fichiers de configuration, et permet l'obtention des fonctionnalits suivantes :

**compatibilit** : support UTF-8 pour la base `template1`

**rles** : cration de deux rles (groupes) distincts permettant l'un la lecture et l'autre criture ; ils seront hrits par les utilisateurs crs manuellement `posteriori`

**cration** : cration de la base qui recevra les donnes

**scurit** : ajout d'un mot de passe pour l'utilisateur `postgres`

**journalisation** : connexions et dconnexions

**authentification** dfinition de la politique de securit

**connexions** : autorisation des connexions `distance`

#### 3.2 Prparation de l'environnement d'excution

Le rle de ce script est de prparer l'environnement d'excution. Sa tche principale est d'effectuer une installation de `miniconda` et d'importer un l'environnement adquat. Cet environnement aura t pralablement export et conserv dans l'arborescence du projet.

## 4 Applicatif Python

### 4.1 Description fonctionnelle

Le code `Python` devra fournir les fonctionnalités suivantes :

- configuration complète via fichiers de configuration
- journalisation des actions effectuées
- téléchargement des données
- décompression des données
- conversion des données
- création de la base
- import des données dans la base
- interrogation de la base

### 4.2 Modularité, découpage en sous-ensembles

Le découpage en sous-ensembles s'articule autour de modules. Chaque module regroupe un ensemble logique de fonctionnalités. La figure suivante montre l'arbre décrivant une structure possible du projet.

[height=15em]arborescence

Cette structure n'est pas définitive. En effet, elle est sujette à être modifiée ou redécoupée si nécessaire. Cependant, le découpage montrera toujours clairement les différentes tâches, de manière à les séparer. En effet les modules de plus haut niveau correspondent aux tâches principales de l'application.