



# Cahier des charges

Stage de fin de DUT à l'Institut Français des Sciences et Technologies des Transports, de l'Aménagement  
et des Réseaux

IUT Informatique de Nantes

Rémi TAUNAY

5 mai 2017

# Table des matières

<b>I</b>	<b>Présentation générale du problème</b>	<b>3</b>
<b>1</b>	<b>Projet</b>	<b>3</b>
1.1	Finalité . . . . .	3
1.2	Problématique . . . . .	3
1.3	Énoncé du besoin . . . . .	3
1.4	Solution apportée . . . . .	4
1.5	Planification . . . . .	4
<b>2</b>	<b>Contexte</b>	<b>4</b>
2.1	Études déjà effectuées . . . . .	4
2.2	Nature des prestations demandées . . . . .	5
2.3	Parties concernées par le déroulement du projet et ses résultats . . . . .	5
<b>II</b>	<b>Détail technique du besoin</b>	<b>5</b>
<b>3</b>	<b>Étapes préliminaires : scripts shell</b>	<b>5</b>
3.1	Installation et configuration de PostgreSQL . . . . .	5
3.2	Préparation de l'environnement d'exécution . . . . .	6
<b>4</b>	<b>Applicatif Python</b>	<b>6</b>
4.1	Description fonctionnelle . . . . .	6
4.2	Modularité, découpage en sous-ensembles . . . . .	6

## Première partie

# Présentation générale du problème

## 1 Projet

### 1.1 Finalité

L'objectif de ce projet est de créer une base de données géo-référencées. Cette base est destinée à constituer un rassemblement volumineux de données géographiques sur le territoire. Les données sont rapatriées de sources diverses puis traitées automatiquement. Enfin, elle 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 Énoncé 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 rapatriant 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 énoncés ci-dessous.

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

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

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

L'ensemble de ces spécifications s'effectueront à 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'altérer le code source pour contrôler le comportement du logiciel.

De plus, seuls les paramètres absolument nécessaires seront à préciser, comme par exemple la structure de la base si l'utilisateur souhaite la créer. L'utilisateur aura bien entendu la liberté de redéfinir les autres paramètres au cas où leurs valeurs par défaut ne lui conviennent pas.

## 1.4 Solution apportée

Pour répondre au besoin énoncé ci-dessus, une base de données centralisée ainsi que des procédures d'importation, de nettoyage, et de traitement de données sont créés. L'ensemble de ces solutions permet notamment :

- un stockage massif performant de données
- l'élimination des tâches de pré-traitement : récupération, nettoyage, uniformisation, recherche de liens (clés & relations)
- la mutualisation des données, rendues accessibles à de multiples utilisateurs
- la mise à disposition de requêtes basiques de sélection avec application de filtres (attributaires, géographiques)

## 1.5 Planification

Diagramme de Gantt :

	15	16	17	18	19	20	21	22	23	24
.....sujet										
exploration du sujet, discussions										
rédaction de documents ressources										
rédaction du cahier des charges										
rédaction du rapport de stage										
.....bash										
script d'installation PostgreSQL										
script de configuration de l'environnement										
.....python										
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										

## 2 Contexte

### 2.1 Études déjà effectuées

Marc NOËL a d'ores-et-déjà effectué un travail préliminaire sur les données et établi les liens existants. En effet, en parcourant et effectuant des traitements sur ces données il a pu concevoir le schéma de la base de données.

C'est ce schéma qu'il revient ici d'implémenter, en discutant des adaptations mineures éventuelles. Ce

schéma se veut proche des données d'entrée. De cette manière, l'intégration des données est facilitée.

## 2.2 Nature des prestations demandées

En premier lieu, le développement de deux scripts shell (`bash`). Ces scripts, non interactifs, vont venir préparer l'exécution ultérieure 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 préparer l'exécution du code `Python` en installant les dépendances (les packages) auxquels il fait appel.

En second lieu, le développement d'un script `Python` chargé de manipuler les données. Ce dernier doit effectuer un ensemble de traitements tels que précisés par l'utilisateur.

## 2.3 Parties concernées par le déroulement du projet et ses résultats

En ce qui concerne l'IUT informatique de Nantes, Monsieur Loïc JEZEQUEL, est chargé en tant qu'enseignant référent de suivre et évaluer ce stage. Ce suivi est réalisé via un message électronique envoyé en chaque fin de semaine, ainsi qu'une visite sur le lieu de stage. De même, plusieurs rendus sont à fournir tel qu'un résumé 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 rôle de tuteurs. Ils seront également les commanditaires et les clients du développement effectué.

Enfin, une soutenance ponctue le stage. Le jury sera alors composé de Pierre HANKACH, Loïc JEZEQUEL, ainsi qu'un second professeur de l'IUT.

## Deuxième partie

# Détail technique du besoin

## 3 Étapes préliminaires : scripts shell

### 3.1 Installation et configuration de PostgreSQL

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

En second lieu, ce script a pour rôle de configurer `PostgreSQL`. Cela est réalisé via l'édition automatisée de fichiers de configuration, et permet l'obtention des fonctionnalités suivantes :

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

**rôles** : création de deux rôles (groupes) distincts permettant l'un la lecture et l'autre écriture ; ils seront hérités par les utilisateurs créés manuellement à posteriori

**création** : création de la base qui recevra les données

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

**journalisation** : connexions et déconnexions

**authentification** définition de la politique de sécurité

**connexions** : autorisation des connexions à distance

## 3.2 Préparation de l'environnement d'exécution

Le rôle de ce script est de préparer l'environnement d'exécution. Sa tâche principale est d'effectuer une installation de `miniconda` et d'importer un l'environnement adéquat. Cet environnement aura été préalablement exporté et conservé dans l'arborescence du projet.

# 4 Applicatif Python

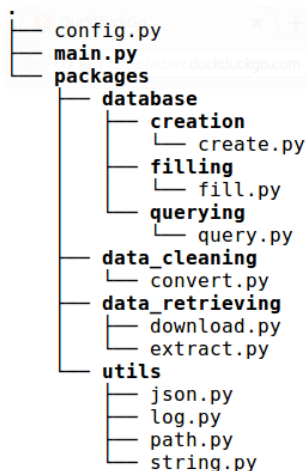
## 4.1 Description fonctionnelle

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.



Cette structure n'est pas définitive. En effet, elle est sujette à être étoffée 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.