



**Stagiaire :** Basile André

**Tuteur entreprise :** Rémi Bouilly

**Tuteur académique :** Jean-François Favennec

DEAL Réunion  
Service Eau et Biodiversité  
Rémi BOUILLY

A handwritten signature in black ink, appearing to read "Bouilly".

BRETAGNE  
**INP** Enib  
UBO

b1andre@enib.fr

**Basile André**

# Remerciements

Je tiens tout d'abord à remercier mon tuteur en entreprise, M. Rémi Bouilly, pour son accompagnement tout au long de ce stage. Il m'a accordé sa confiance dès le début du projet et m'a permis de travailler avec une grande autonomie. Sa disponibilité et la qualité de ses retours ont été essentielles pour structurer mon travail et orienter mes choix techniques. Nos échanges réguliers ont constitué un appui précieux, tant pour résoudre des problématiques concrètes que pour mieux comprendre les enjeux liés aux données naturalistes et à leur valorisation.

Je remercie également mon tuteur académique, M. Jean-François Favennec, pour son suivi et ses conseils tout au long du stage. Ses remarques m'ont aidé à prendre du recul sur les travaux réalisés et à adopter une démarche plus rigoureuse et structurée. Son accompagnement a contribué à inscrire ce projet dans une véritable logique d'ingénierie.

Je souhaite remercier l'ensemble des agents de la DEAL Réunion pour leur accueil et leur bienveillance. J'ai évolué dans un environnement de travail agréable, où les échanges étaient simples et constructifs. Les discussions avec les différents agents m'ont permis de mieux appréhender le fonctionnement d'une administration publique et les contraintes associées à la gestion des données environnementales.

Je remercie tout particulièrement les membres de l'Unité Biodiversité (UBIO) pour leur disponibilité et leur intérêt pour le travail mené. Leurs retours concrets et leur expertise métier ont donné du sens aux développements réalisés. Ils ont contribué à orienter le projet vers des solutions réellement utiles et adaptées aux besoins du service.

Enfin, je remercie l'ENIB pour la formation d'ingénieur généraliste dispensée, qui m'a permis de mobiliser des compétences variées tout au long de ce stage. Cette expérience a été particulièrement enrichissante, tant sur le plan technique que personnel. Elle a renforcé mon intérêt pour le développement d'outils numériques ayant un impact concret, en lien avec les enjeux environnementaux et territoriaux.

# Résumé (FR)

Au cours de ma formation d'ingénieur généraliste à l'ENIB (École Nationale d'Ingénieurs de Brest), j'ai eu l'opportunité d'explorer des domaines variés, dont l'informatique, le traitement des données et la communication réseau. Ces enseignements, combinés au contexte environnemental actuel, m'ont donné l'envie d'explorer comment les outils numériques peuvent répondre à des enjeux concrets, en particulier ceux liés à l'environnement et au milieu marin, un domaine qui m'inspire depuis longtemps. C'est cette motivation qui m'a conduit à choisir la DEAL (Direction de l'Environnement, de l'Aménagement et du Logement), où le développement d'outils informatiques joue un rôle clé dans la valorisation et l'accessibilité des données, offrant une occasion idéale d'allier mes compétences techniques à un sujet qui a du sens.

J'ai effectué mon stage au sein de la DEAL Réunion, plus précisément au SEB (Service Eau et Biodiversité), à l'UBIO (Unité Biodiversité). L'île de La Réunion présente des enjeux importants de suivi et de préservation de la biodiversité, en raison de la singularité de ses écosystèmes et des pressions qu'ils subissent (notamment les EEE (Espèces Exotiques Envahissantes) et le développement du territoire). Ce contexte renforce la nécessité d'un suivi rigoureux des données naturalistes.

La DEAL utilise Géonature (Application de gestion et de centralisation des données naturalistes) pour centraliser et gérer ces données. Le sujet principal de mon stage consistait à développer un module externe permettant d'y importer automatiquement les données marines issues de Quadrige (Système d'information de l'Ifremer dédié aux données environnementales marines). Ces données, essentielles pour le suivi environnemental littoral, représentent un enjeu majeur pour les missions de la DEAL.

N'ayant pas suivi le module CAI (Conception d'Applications Interactives), ce stage m'a permis d'acquérir des compétences complémentaires en développement logiciel et applicatif, à travers la conception d'un module d'import, la manipulation de données et l'interfaçage de systèmes hétérogènes, afin d'obtenir un outil simple d'utilisation, robuste et maintenable.

Il m'a également offert une immersion dans le fonctionnement d'une administration publique de taille moyenne, où l'alternance entre travail autonome et collaboration avec les agents en charge des données naturalistes et de l'administration de Géonature s'est révélée particulièrement enrichissante. Cette expérience en environnement professionnel réel a renforcé ma compréhension

---

des enjeux liés à la gestion des données environnementales et a affiné ma capacité à questionner la pertinence, l'utilité et le sens des projets auxquels je contribue, afin de m'assurer qu'ils soient en accord avec mes valeurs et porteurs d'un impact positif.

# Abstract (EN)

As part of my general engineering studies at ENIB, I had the opportunity to explore a wide range of fields, including computer science,

data processing and network communications. These courses, combined with the current environmental context, strengthened my interest in understanding how digital tools can address real-world challenges — particularly those related to the environment and the marine domain, which has long inspired me. This motivation led me to choose DEAL, where the development of digital tools plays a key role in enhancing the value and accessibility of environmental data, offering an ideal opportunity to align my technical skills with a topic that holds meaning for me.

I completed my internship at DEAL Réunion, more specifically within the SEB and the UBIO. Réunion Island faces major biodiversity challenges due to its high proportion of endemic species and the presence of particularly sensitive natural habitats. These ecosystems are under strong pressure, notably from EEE and ongoing land-use development, which reinforces the need for rigorous naturalist data. DEAL uses Géonature to centralise and manage these datasets. The main objective of my internship was to develop an external module for this application, enabling the automated import of marine and coastal environmental data produced by Ifremer (Institut Français de Recherche pour l'Exploitation de la Mer). These data are essential for biodiversity monitoring and are accessed through Quadrige, Ifremer's dedicated information system.

Since I had not taken the CAI module, this internship allowed me to acquire complementary skills in software and application development through the design of an import module, data manipulation and the interfacing of heterogeneous systems, with the aim of producing a simple, robust and maintainable tool.

It also offered me insight into the functioning of a medium-sized public administration, where the alternation between autonomous work and collaboration with staff in charge of naturalist data and the administration of Géonature proved particularly enriching. This experience in a real professional environment strengthened my understanding of the challenges associated with environmental data management and refined my ability to reflect on the relevance, usefulness and purpose of the projects I contribute to, ensuring that they align with my values and have a positive impact.

# Table des matières

<b>Remerciements</b>	<b>1</b>
<b>Résumé (FR)</b>	<b>2</b>
<b>Abstract (EN)</b>	<b>4</b>
<b>Glossaire</b>	<b>7</b>
<b>1 Présentation de l'entreprise</b>	<b>9</b>
1.1 DEAL Réunion . . . . .	9
1.2 Diagnostic RSE . . . . .	10
1.2.1 Risques et impacts des activités . . . . .	10
1.2.2 Stratégie globale en matière de RSE . . . . .	11
1.2.3 Conclusion personnelle . . . . .	11
<b>2 Organisation du stage</b>	<b>13</b>
2.1 Contexte du stage . . . . .	13
2.2 Objectifs du stage . . . . .	13
2.3 Méthodologie de travail . . . . .	14
<b>3 Introduction à GeoNature</b>	<b>17</b>
3.1 Principe général de GeoNature . . . . .	17
3.2 Architecture technique . . . . .	18
3.3 Organisation interne et modules . . . . .	18
<b>4 Développement des modules d'imports de données</b>	<b>19</b>
4.1 Contexte du projet . . . . .	19
4.2 Périmètre fonctionnel . . . . .	21
4.3 Conception technique du module d'import Quadrige . . . . .	21
4.3.1 Architecture générale et découplage frontend/backend . . . . .	22
4.3.2 Gestion de la configuration et des paramètres sensibles . . . . .	22
4.3.3 Extraction des programmes Quadrige . . . . .	22
4.3.4 Extraction des données et orchestration des traitements asynchrones . . . . .	24
4.3.5 Gestion des états, des erreurs et de la traçabilité . . . . .	24
4.3.6 Interface utilisateur et pilotage des extractions . . . . .	26
4.3.7 Synthèse . . . . .	28
4.4 Contraintes, dépendances et livrables du module Quadrige . . . . .	28
4.5 Extension du module api2GN pour l'intégration des données Pl@ntNet . . . . .	28
4.5.1 Rôle du module api2GN (Module Géonature permettant la création et l'utilisation de parseurs) dans l'écosystème GeoNature . . . . .	28
4.5.2 Objectifs et positionnement du parser Pl@ntNet . . . . .	29
4.5.3 Architecture technique et principes de conception . . . . .	30
4.5.4 Configuration dynamique et pilotage par fichier TOML . . . . .	30
4.5.5 Résolution taxonomique et contrôle de la qualité des données . . . . .	32
4.5.6 Traçabilité, sécurité et automatisation des imports . . . . .	33
4.5.7 Apports du travail réalisé . . . . .	34
<b>5 Travaux réalisés et compétences acquises</b>	<b>35</b>
5.1 Synthèse des travaux réalisés . . . . .	35
5.1.1 Vue d'ensemble des développements . . . . .	35
5.1.2 Travaux principaux : module Quadrige . . . . .	36

5.1.3	Travaux complémentaires : Pl@ntNet (Application mobile de reconnaissance des plantes par photo) . . . . .	36
5.2	Positionnement et choix d'ingénierie . . . . .	37
5.3	Arbitrages techniques . . . . .	37
5.4	Difficultés rencontrées et solutions apportées . . . . .	38
5.5	Compétences acquises et prise de recul . . . . .	40
<b>6</b>	<b>Perspectives</b>	<b>42</b>
6.1	Perspectives d'évolution du module d'import Quadrige . . . . .	42
6.1.1	Amélioration de l'ergonomie et de l'intégration visuelle . . . . .	42
6.1.2	Ajout d'une étape de mise en forme et d'intégration des données . . . . .	43
6.1.3	Clarification et extension de la gestion de la configuration . . . . .	43
6.1.4	Vers une automatisation partielle et maîtrisée des imports . . . . .	44
6.1.5	Renforcement de la stratégie de tests et de validation . . . . .	44
6.2	Perspectives autour du parser Pl@ntNet et du module api2gn . . . . .	45
6.2.1	Ajout d'une interface d'administration dédiée . . . . .	45
6.2.2	Contraintes liées au module api2gn . . . . .	45
6.2.3	Renforcement des mécanismes de validation et d'incrémentalité . . . . .	45
6.3	Ouvertures transversales . . . . .	46
<b>7</b>	<b>Conclusion</b>	<b>47</b>
<b>Annexes</b>		<b>48</b>
<b>Webographie</b>		<b>50</b>

# Glossaire

<b>API</b>	Interface de Programmation de l'Application
<b>API REST</b>	Interface de programmation reposant sur le protocole HTTP, utilisant des opérations standard (GET, POST, PUT, DELETE) pour accéder et manipuler des ressources identifiées par des URL
<b>api2GN</b>	Module Géonature permettant la création et l'utilisation de parseurs
<b>Borbonica</b>	Plateforme régionale de diffusion des données naturalistes du SINP
<b>CAI</b>	Conception d'Applications Interactives
<b>CSV</b>	Comma-Separated Values, format de fichier texte structuré permettant de représenter des données tabulaires
<b>DEAL</b>	Direction de l'Environnement, de l'Aménagement et du Logement
<b>EEE</b>	Espèces Exotiques Envahissantes
<b>ENIB</b>	École Nationale d'Ingénieurs de Brest
<b>ERC</b>	Éviter – Réduire – Compenser
<b>GraphQL</b>	Langage de requête permettant d'interroger une API en décrivant précisément la structure des données attendues
<b>Géonature</b>	Application de gestion et de centralisation des données naturalistes
<b>Ifremer</b>	Institut Français de Recherche pour l'Exploitation de la Mer
<b>JSON</b>	JavaScript Object Notation, format léger d'échange de données structuré et lisible

<b>MNHN</b>	Muséum national d'Histoire naturelle
<b>OFB</b>	Office Français de la Biodiversité
<b>Pl@ntNet</b>	Application mobile de reconnaissance des plantes par photo
<b>PNRun</b>	Parc National de La Réunion
<b>Quadrigé</b>	Système d'information de l'Ifremer dédié aux données environnementales marines
<b>RNE</b>	Responsabilité Numérique des Entreprises
<b>RSE</b>	Responsabilité Sociétale des Entreprises
<b>SEB</b>	Service Eau et Biodiversité
<b>SEOR</b>	Société d'Études Ornithologiques de La Réunion
<b>SINP-974</b>	Système d'Information de l'iNventaire du Patrimoine naturel de la Réunion
<b>SREPEN</b>	Société réunionnaise pour l'étude et la protection de la nature
<b>TAXREF</b>	Référentiel taxonomique national de la faune, de la flore et de la fonge, maintenu par le MNHN
<b>TAXREF-LD</b>	Service web du MNHN exposant le référentiel taxonomique TAXREF sous forme de données liées (Linked Data), permettant l'interrogation et la résolution taxonomique via une API
<b>UBIO</b>	Unité Biodiversité

# 1. Présentation de l'entreprise

## 1.1. DEAL Réunion

La Direction de l’Environnement, de l’Aménagement et du Logement (DEAL) est le service déconcentré de l’État. Elle met en œuvre, à l’échelle régionale, les politiques publiques relevant du Ministère de la Transition Écologique et de la Cohésion des Territoires, ainsi que du Ministère de la Transition Énergétique. La Réunion est un territoire insulaire soumis à de fortes pressions environnementales et à des enjeux d’aménagement complexes. Dans ce contexte, la DEAL occupe une place centrale. Elle se situe au croisement des questions d’environnement, de biodiversité, d’eau, d’urbanisme et de développement territorial.

La DEAL Réunion assure l’application des réglementations environnementales. Elle instruit les projets d’aménagement et gère les risques naturels. Elle suit la ressource en eau et met en œuvre les politiques de protection des milieux naturels. Elle travaille en étroite collaboration avec les collectivités et plusieurs établissements publics, tels que l’OFB (Office Français de la Biodiversité), le PNRun (Parc National de La Réunion) ou l’Ifremer. Elle coopère aussi avec de nombreuses associations locales, comme la SEOR (Société d’Études Ornithologiques de La Réunion) ou la SREPEN (Société réunionnaise pour l’étude et la protection de la nature), qui contribuent activement au suivi et à la préservation de la biodiversité insulaire.

Au sein de cette structure, le Service Eau et Biodiversité (SEB) porte les missions liées à la préservation des milieux aquatiques et terrestres. Il développe la connaissance des espèces, veille à leur protection et régule les activités susceptibles d’impacter la biodiversité. Le SEB se trouve ainsi au cœur des enjeux écologiques de l’île.

Mon stage s’est déroulé au sein de l’Unité Biodiversité (UBIO). Cette unité est responsable du suivi des espèces et des habitats naturels. Elle gère et valorise les données naturalistes et instruit les dossiers réglementaires liés à la biodiversité. Elle anime également le Système d’Information sur la Nature et les Paysages (SINP-974 (Système d’Information de l’iNventaire du Patrimoine naturel de la Réunion)) régional et assure la gestion de la plateforme Borbonica. L’unité intervient aussi sur des thématiques transversales telles que les espèces exotiques envahissantes (EEE), la séquence ERC (Éviter – Réduire – Compenser) ou la diffusion des connaissances naturalistes. L’organisation interne de l’UBIO est présentée en annexe (Fig. 7.1).

L’organigramme interne montre une équipe composée de profils scientifiques, techniques et administratifs. Ces personnels travaillent de manière complémentaire pour répondre aux enjeux liés à la biodiversité du territoire. Mon stage s’inscrit dans cette dynamique, au sein du pôle dédié aux données naturalistes. Il contribue à la structuration et à la modernisation des outils numériques utilisés par la DEAL.

Cette présentation de la DEAL et de son organisation permet de situer le contexte global de mon stage. Le chapitre suivant propose un **diagnostic RSE (Responsabilité Sociétale des**

**Entreprises)** de la structure. Il vise à évaluer ses pratiques au regard des enjeux sociaux, environnementaux et organisationnels.

## 1.2. Diagnostic RSE

En raison des responsabilités qui lui sont confiées, la DEAL Réunion doit intégrer des considérations sociales, environnementales et organisationnelles dans l'ensemble de ses pratiques. Le diagnostic qui suit s'appuie sur la norme ISO 26000 et sur les principes de la RNE (Responsabilité Numérique des Entreprises), afin d'évaluer la manière dont la structure prend en compte ces enjeux.

### 1.2.1. Risques et impacts des activités

#### Impacts environnementaux

La DEAL n'engendre pas d'impacts industriels directs. Cependant, plusieurs de ses activités présentent des risques environnementaux indirects :

- un usage intensif d'outils numériques (bases de données naturalistes, SIG, plateformes SINP-974, Géonature, Borbonica (Plateforme régionale de diffusion des données naturalistes du SINP)), impliquant une consommation énergétique importante et une dépendance aux infrastructures serveurs ;
- le stockage et le traitement de volumes croissants de données, contribuant à l'empreinte carbone numérique et à la production de déchets électroniques liés au renouvellement du matériel ;
- des déplacements professionnels réguliers pour l'instruction de projets, le suivi de chantiers ou les missions de terrain, occasionnant des émissions de gaz à effet de serre.

Ces activités consomment de l'énergie, mobilisent des ressources non renouvelables et contribuent à l'empreinte environnementale globale du service.

#### Impacts sociétaux

Les missions de la DEAL comportent également plusieurs risques sociétaux :

- une influence directe sur les décisions d'aménagement (urbanisme, infrastructures, gestion des risques), pouvant affecter l'organisation du territoire et la qualité de vie des habitants.
- une responsabilité dans l'instruction des permis CITES et le suivi des espèces menacées, dont une mauvaise évaluation pourrait fragiliser le patrimoine naturel.
- la coordination des actions de lutte contre les espèces exotiques envahissantes (EEE), où un retard ou un défaut d'intervention peut entraîner une dégradation durable des écosystèmes.
- une forte dépendance aux partenaires (collectivités, OFB, Ifremer, PNRun, associations), rendant certaines actions vulnérables à des manques de coordination ou de moyens.

---

Ces risques peuvent affecter la cohérence des politiques publiques, la préservation des milieux naturels et l'équité territoriale.

### **Impacts sociaux**

Le fonctionnement interne de la DEAL comporte également des risques sociaux. Ses équipes travaillent dans un environnement pluridisciplinaire soumis à des contraintes organisationnelles et réglementaires :

- une charge de travail variable selon les périodes, pouvant entraîner des tensions dans la répartition des tâches et la disponibilité des agents.
- des missions de terrain présentant des risques pour la santé et la sécurité (déplacements fréquents, conditions difficiles, chantiers, zones accidentées).
- un besoin constant de mise à jour des compétences (réglementations, outils numériques, SIG), créant un risque d'obsolescence professionnelle pour les agents non formés.
- un renouvellement régulier des stagiaires ou contractuels, pouvant fragiliser la continuité des projets ou l'organisation des équipes.

Ces éléments peuvent affecter la qualité de vie au travail, la sécurité en mission et la stabilité organisationnelle.

#### **1.2.2. Stratégie globale en matière de RSE**

Bien qu'elle agisse dans un cadre réglementaire strict, la DEAL cherche à renforcer la cohérence et la qualité de ses pratiques internes. Elle s'inscrit dans une dynamique de *pré-conformité active*, allant au-delà des obligations minimales en matière d'organisation, de gestion de l'information et de modernisation numérique. Cette démarche vise à améliorer la fiabilité des données environnementales, à harmoniser les outils utilisés par les services et à garantir une application rigoureuse des réglementations.

Sur le plan sociétal, la DEAL renforce ses méthodes de travail pour limiter les risques identifiés. Elle veille à la transparence dans l'instruction des projets d'aménagement, à sécuriser les procédures liées aux permis CITES et à structurer davantage la coordination des actions de lutte contre les (EEE).

Cette dynamique repose également sur une coopération accrue avec les collectivités territoriales, les établissements publics et les associations naturalistes, afin de réduire les vulnérabilités liées à la gouvernance et à la circulation de l'information. Ces partenariats contribuent à une meilleure gestion des risques sociaux, environnementaux et sociaux auxquels le territoire est confronté.

#### **1.2.3. Conclusion personnelle**

Ce diagnostic montre que la DEAL Réunion intègre progressivement les enjeux de la RSE et de la RNE au cœur de ses pratiques, malgré les contraintes propres à une administration publique. Cette approche illustre la volonté de la structure de moderniser son fonctionnement

---

et de réduire ses vulnérabilités, tout en répondant aux attentes réglementaires et territoriales. Cette réflexion renforce l'idée que le numérique et l'organisation interne jouent un rôle essentiel dans la transition écologique et l'amélioration des politiques publiques.

## 2. Organisation du stage

### 2.1. Contexte du stage

Le SINP-974 est un dispositif collectif de mise en partage des données d'observations d'espèces sauvages sur l'île. Il a été mis en service à La Réunion en 2018, au travers de la plateforme Borbonica gérée conjointement par PNRUn et la DEAL.

Borbonica s'appuie sur un portail web, qui permet d'accéder aux différentes interfaces utilisateurs du SINP :

- Borbonica obs : consultation cartographique des données du SINP. Il s'agit d'une interface web basée sur le plugin Lizmap et reposant sur une base de données PostgreSQL. Cette plateforme offre un accès aux données selon différents niveaux et profils (grand public, experts, validateurs, etc.) ;
- Borbonica atlas : consultation des fiches espèces disponibles dans le SINP, sous forme de synthèse des principales informations (observations, documents, photos, etc.). Il s'agit d'un outil issu de la solution open source GéoNature ;
- Borbonica stats : module statistique qui permet de consulter des tableaux de bord chiffrés sur le contenu de Borbonica (données disponibles, principaux usages, etc.). Ce module s'appuie sur un projet Lizmap, exploitant des données de la base PostgreSQL.

Afin de pérenniser le SINP 974, un projet a été lancé en 2023, avec pour principaux objectifs la modernisation et la simplification du fonctionnement du système. Plusieurs nouvelles fonctionnalités vont être ajoutées, et certaines briques logicielles seront remplacées par de nouvelles solutions plus modernes et plus simples à utiliser et à maintenir. C'est le cas notamment de la solution Génature qui est en cours de déploiement pour remplacer le périmètre de Borbonica obs.

### 2.2. Objectifs du stage

À la suite de la modernisation engagée pour le SINP-974 et du déploiement progressif de GeoNature au sein de la DEAL, le stage avait pour finalité d'apporter un appui opérationnel au renforcement de la chaîne d'acquisition et de gestion des données naturalistes. L'objectif principal était de faciliter l'intégration de données externes dans GeoNature et, plus largement, d'enrichir les informations accessibles via la plateforme régionale Borbonica.

Dans ce cadre, plusieurs objectifs spécifiques ont été définis :

- développer un module externe GeoNature interfacé avec le système d'information Qua-drige de l'Ifremer, afin d'automatiser la récupération, le filtrage et la préparation des données issues du milieu marin ;
  - poser les fondations techniques nécessaires à une future intégration des données produites par l'application Pl@ntNet, en réalisant un client Python et un premier outil de structu-ration des résultats ;
  - documenter les procédures d'appel aux API et proposer une organisation des dévellope-ments garantissant leur maintenabilité et leur évolution dans le temps.

En complément du module Quadrige, une seconde orientation du stage concernait la préparation d'un futur connecteur GeoNature dédié aux données issues de Pl@ntNet. Pl@ntNet produit chaque jour un volume important d'observations naturalistes géolocalisées, particulièrement pertinent pour le suivi des EEE végétales et des dynamiques de végétation. Afin d'explorer la faisabilité de cette intégration, un travail préliminaire a consisté à :

- étudier la documentation de l'API Pl@ntNet v3,
  - concevoir un client Python robuste, capable d'interroger l'API selon plusieurs critères (taxon, polygone GeoJSON, plage temporelle),
  - structurer les données extraites au format *Darwin Core* afin d'assurer une compatibilité immédiate avec GeoNature et le SINP-974.

Ce travail prépare le développement d'un parser complet destiné au module `api2gn` de GeoNature.

Ces objectifs s'inscrivent pleinement dans la stratégie portée conjointement par la DEAL Réunion et le PNRun, visant à fiabiliser et à harmoniser les données du SINP-974 tout en renforçant la représentation de thématiques encore peu renseignées, comme le milieu marin. Ils ont guidé l'ensemble des actions menées durant le stage et constituent le cadre des travaux détaillés dans les sections suivantes.

### 2.3. Méthodologie de travail

La méthodologie adoptée au cours du stage s'est construite de manière progressive et pragmatique. Elle a été définie en tenant compte des contraintes techniques du projet, du cadre institutionnel de la DEAL et de l'organisation particulière de l'équipe. Le projet étant mené uniquement à deux personnes — mon maître de stage, Rémi Bouilly, et moi-même — il n'était pas pertinent de mettre en place une méthodologie formelle lourde. Le cadre de travail retenu s'inspire néanmoins des principes des méthodes agiles, tout en restant volontairement souple.

Le développement du module Quadrige et des outils associés a ainsi été conduit selon une démarche itérative et incrémentale. Le travail était organisé en cycles courts, assimilables à des sprints d'une durée moyenne d'environ une semaine. Chaque sprint correspondait à un ensemble cohérent de fonctionnalités, définies en amont, puis développées, testées et ajustées

avant d'engager le cycle suivant. Cette organisation permettait d'obtenir régulièrement des résultats fonctionnels, même partiels, tout en identifiant rapidement les difficultés techniques ou fonctionnelles.

Dans ce contexte de travail en effectif réduit, il n'était ni nécessaire ni utile de mettre en place des rituels agiles classiques tels que les daily meetings ou les revues de sprint formelles. En revanche, des points réguliers étaient organisés une à deux fois par semaine avec mon maître de stage. Ces échanges permettaient de faire un état d'avancement, de valider les choix techniques effectués et d'ajuster les priorités à court terme en fonction des besoins métier. Ils ont joué un rôle central dans le bon déroulement du projet, en assurant une compréhension partagée des objectifs tout en me laissant une large autonomie dans la mise en œuvre.

Le cadrage initial du stage s'est appuyé sur deux cahiers des charges distincts, fournis par mon maître de stage. Le premier concernait le développement du module externe GeoNature interfacé avec Quadrige, tandis que le second portait sur la conception d'un parser dédié aux données issues de Pl@ntNet et à leur structuration au format Darwin Core. Ces documents définissaient le périmètre fonctionnel attendu, les principales contraintes techniques ainsi que les livrables envisagés. Ils ont servi de fil conducteur tout au long du stage, tout en restant suffisamment flexibles pour intégrer des ajustements liés aux contraintes rencontrées ou à l'évolution du projet.

Afin de suivre l'avancement global du projet et de visualiser les grandes phases du stage, un diagramme de Gantt a été utilisé.

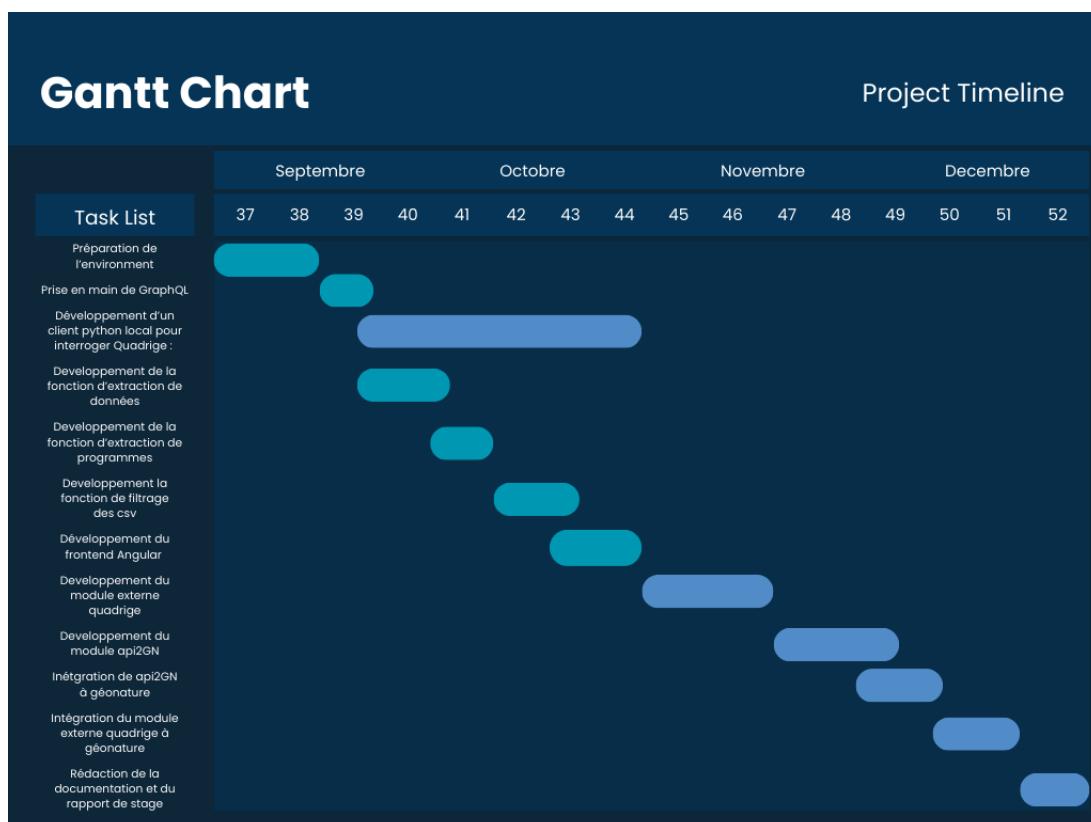


Figure 2.1: Diagramme de Gantt du déroulement du stage

Il a permis de planifier les principales étapes, depuis la prise en main de l'environnement GeoNature jusqu'aux phases de tests, de documentation et de préparation des livrables. Cet outil offrait une vision synthétique du projet et facilitait l'anticipation des périodes plus sensibles, notamment lors des phases de migration des serveurs ou d'attente d'accès aux environnements distants.

À un niveau plus opérationnel, le travail quotidien était organisé sous forme de tâches unitaires, réparties selon un statut simple : à faire, en cours ou terminées. Cette organisation, inspirée des tableaux de type Kanban, permettait de visualiser rapidement l'état d'avancement, de prioriser les actions et de conserver une trace claire du travail réalisé. Elle s'est révélée particulièrement adaptée à un projet individuel encadré, en apportant une structure légère sans alourdir inutilement le processus.

La gestion du code source reposait intégralement sur l'utilisation de Git. L'ensemble des développements a été versionné dès les premières phases du projet, aussi bien pour le module Quadrige que pour les outils liés à Pl@ntNet. Les évolutions étaient enregistrées de manière régulière, avec des messages de commit explicites. Cette pratique a permis de sécuriser les développements, de tester différentes approches sans risque de perte de code et de faciliter le transfert vers les serveurs de la DEAL une fois l'accès ouvert. Dans un contexte où l'accès aux environnements distants était limité et parfois instable, Git a également servi de principal moyen de synchronisation entre l'environnement local et l'environnement distant.

Le développement a d'abord été mené intégralement en local, en l'absence d'un accès immédiat aux serveurs du SINP-974. Cette situation a nécessité la mise en place d'un environnement de travail stable, capable de reproduire au mieux les conditions de production. La séparation stricte entre le backend Python/Flask et le frontend Angular, conforme à l'architecture modulaire de GeoNature, a facilité cette organisation et permis de développer chaque composante de manière indépendante.

Les contraintes liées au proxy ministériel et aux versions spécifiques des outils techniques ont également nécessité une attention particulière à la configuration de l'environnement. L'utilisation de pyenv a permis de stabiliser la version de Python utilisée, condition indispensable pour garantir la compatibilité avec GeoNature et ses dépendances.

Enfin, l'ensemble du travail s'est appuyé sur une démarche de validation progressive. Les appels à l'API Quadrige ont d'abord été testés à l'aide de scripts Python indépendants, avant leur intégration dans le module GeoNature. De la même manière, les premières interfaces Angular ont été testées en local afin de vérifier leur cohérence fonctionnelle et leur ergonomie avant toute intégration plus large.

Cette méthodologie, mêlant principes agiles, rigueur technique et adaptation constante au contexte, a permis d'assurer la robustesse des développements malgré un environnement parfois contraint. Elle a également facilité l'anticipation des étapes futures d'intégration et de reprise du projet par la DEAL ou ses partenaires.

## 3. Introduction à GeoNature

### 3.1. Principe général de GeoNature

GeoNature est une application web dédiée à la gestion, la centralisation et la valorisation des données naturalistes. Développé initialement en 2010 puis entièrement refondu en 2017, le projet est aujourd’hui maintenu par le PNRun. L’application repose sur une architecture moderne combinant un backend Python/Flask et un frontend Angular, ce qui lui permet d’assurer à la fois la saisie, la consultation, la validation et la restitution des données.

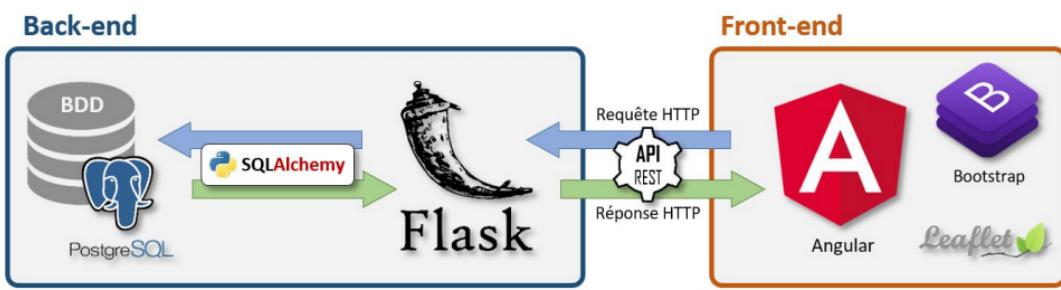


Figure 3.1: Architecture générale du backend et du frontend de GeoNature

Le fonctionnement de GeoNature est modulaire : un noyau applicatif fournit les briques communes (API, référentiels, schéma de synthèse), et différents modules viennent étendre les fonctionnalités selon les besoins (Occtax, Occhab, Validation, Import, Export, etc.). La documentation officielle détaille l’ensemble de cette architecture, ([documentation officielle](#)) ainsi que les sources du projet, disponibles sur [GitHub](#).

### 3.2. Architecture technique

GeoNature combine une partie serveur (backend) et une partie cliente (frontend) qui dialoguent via une API REST.

Le backend est développé en Python à l'aide du framework Flask. Il assure l'ensemble des traitements métiers, l'accès aux données et la gestion des opérations spatiales via PostgreSQL et PostGIS. La gestion des utilisateurs et de leurs permissions repose sur UsersHub, qui fournit l'authentification et le contrôle des accès. L'API exposée par le backend constitue le point central de communication avec l'interface web.

Le frontend, développé en Angular, constitue l'interface visible par l'utilisateur. Il interroge exclusivement l'API du backend pour afficher les formulaires, les cartes, les graphiques et les données attributaires. Des bibliothèques comme Leaflet ou Bootstrap renforcent les fonctionnalités cartographiques et l'ergonomie générale. Cette séparation nette entre backend et frontend garantit la stabilité et la modularité du système, tout en facilitant les évolutions futures.

### 3.3. Organisation interne et modules

L'application est structurée autour de modules fonctionnels s'appuyant tous sur les mêmes référentiels : taxonomie (TaxHub), nomenclatures, utilisateurs (UsersHub) et schéma de synthèse. Les modules principaux — Occtax pour les observations, Occhab pour les habitats, ou encore la Validation — s'intègrent directement au cœur applicatif.

Chaque module possède son propre schéma de base de données, son API et ses composants Angular. Cette organisation modulaire permet de faire évoluer GeoNature, d'ajouter de nouveaux protocoles d'acquisition ou de développer des extensions externes. La documentation décrit précisément les bonnes pratiques et l'architecture à respecter pour développer un [module GeoNature externe](#).

## 4. Développement des modules d'imports de données

### 4.1. Contexte du projet

Le SINP-974, déployé à La Réunion à travers la plateforme Borbonica, constitue le dispositif régional de référence pour la centralisation, la gestion et la diffusion des données naturalistes. Depuis sa mise en service, il a permis de structurer et de valoriser un volume important d'observations, issues de sources multiples et couvrant un large éventail de taxons.

La dynamique de peuplement de la base reflète toutefois les modalités historiques et opérationnelles de production des données naturalistes sur le territoire. Les observations actuellement intégrées concernent majoritairement les milieux terrestres et dulçaquicoles, qui bénéficient de réseaux d'acteurs structurés, de protocoles d'acquisition largement diffusés et d'outils de saisie directement compatibles avec le SINP.

À l'inverse, les données issues du milieu marin demeurent plus faiblement représentées. Les indicateurs de contenu disponibles en 2024 montrent que les observations relatives aux taxons marins ne constituent qu'une part marginale des données référencées. Cette situation ne traduit pas une absence de connaissances ou de suivis en milieu marin. Elle résulte principalement de la complexité des chaînes d'acquisition, de structuration et de diffusion propres aux données marines, historiquement gérées au sein de systèmes d'information spécialisés.

Dans ce contexte, l'enjeu principal n'était pas de refondre les outils existants, mais de renforcer progressivement la représentation des données marines au sein du SINP-974, en s'appuyant sur des sources de données déjà structurées, pérennes et reconnues institutionnellement. Parmi celles-ci, le système d'information Quadrige, maintenu par l'Ifremer, occupe une place centrale. Quadrige regroupe un volume important de données environnementales et biologiques collectées en milieu marin, issues de programmes de surveillance, d'observation et de recherche conduits sur le long terme.

L'ouverture récente d'une API (Interface de Programmation de l'Application) GraphQL (Langage de requête permettant d'interroger une API en décrivant précisément la structure des données attendues) par l'Ifremer constitue à ce titre une évolution majeure. Elle permet une interrogation directe, fine et structurée des données de Quadrige, en ciblant précisément les programmes, les zones géographiques, les périodes temporelles et les types d'observations d'intérêt. Le fonctionnement de l'API Quadrige et les modalités d'extraction des données sont décrits dans la documentation officielle mise à disposition par l'Ifremer [Documentation officielle de l'API Quadrige](#).

Cette évolution ouvre la voie à une exploitation plus systématique de ces données par des acteurs institutionnels, et à leur mobilisation progressive dans le cadre du SINP-974.

C'est dans ce cadre que s'inscrit le projet développé durant le stage. L'objectif n'était pas de réaliser une intégration directe et entièrement automatisée des données Quadrige dans la base GeoNature, mais de concevoir un module externe permettant d'explorer, d'extraire et de préparer ces données de manière structurée et contrôlée. Le module développé interroge l'API Quadrige afin d'identifier les programmes pertinents pour le territoire réunionnais, puis permet d'en extraire les observations associées selon des critères définis par l'utilisateur. Les résultats de ces extractions ne prennent pas la forme de fichiers immédiatement importables dans GeoNature. Pour chaque programme sélectionné, le module génère une archive compressée contenant plusieurs éléments complémentaires : un fichier CSV (Comma-Separated Values, format de fichier texte structuré permettant de représenter des données tabulaires) brut correspondant aux données extraites, un fichier JSON (JavaScript Object Notation, format léger d'échange de données structuré et lisible) décrivant précisément les filtres appliqués lors de l'appel à l'API, ainsi qu'un fichier README documentant le déroulement de l'export et les éventuelles anomalies rencontrées. Le module conserve également un accès aux trois derniers exports réalisés afin d'en faciliter la consultation et la réutilisation.

En complément de ces archives, le module expose également des liens directs vers les fichiers CSV générés lors de la phase d'extraction des programmes. Deux niveaux de fichiers sont distingués :

- un CSV brut issu directement de la réponse de l'API Quadrige, contenant l'ensemble des instances des programmes dont au moins une occurrence est localisée sur le territoire ciblé ;
- un CSV filtré géographiquement, obtenu par un traitement a posteriori à l'aide de la bibliothèque pandas, permettant d'exclure les instances dont le code de monitoring location ne correspond pas au territoire d'intérêt (par exemple, codes ne commençant pas par le préfixe « 126- » pour La Réunion).

Ce filtrage complémentaire est nécessaire car, lors de l'extraction des programmes, la requête adressée à l'API Quadrige sélectionne l'ensemble des programmes possédant au moins une instance sur le territoire demandé, mais retourne également les autres instances associées à ces programmes, y compris celles situées hors du périmètre géographique ciblé.

Cette approche intermédiaire répond à plusieurs objectifs. Elle garantit d'une part la traçabilité complète des extractions réalisées depuis Quadrige, en conservant une description explicite des paramètres et traitements appliqués. Elle offre d'autre part une souplesse d'usage, en laissant aux administrateurs et gestionnaires de données la possibilité de contrôler, d'analyser et, le cas échéant, d'adapter les fichiers produits avant toute intégration dans GeoNature ou Borbonica.

Ce développement s'inscrit pleinement dans la dynamique d'amélioration continue portée par la DEAL Réunion et ses partenaires. Il vise à renforcer progressivement la place des données issues du milieu marin au sein du système régional, tout en respectant les contraintes techniques, méthodologiques et organisationnelles propres aux outils existants.

## 4.2. Périmètre fonctionnel

Le module d'import devait couvrir l'ensemble de la chaîne d'acquisition : de la découverte des programmes Quadrige jusqu'à la production d'un fichier structuré pour GeoNature. La première étape consistait à interroger l'API en mode authentifié afin d'obtenir la liste des programmes disponibles pour un utilisateur donné. Un filtrage automatique sur un périmètre géographique — principalement La Réunion, mais extensible à d'autres territoires comme les Îles Éparses — permettait d'isoler les programmes pertinents. Une interface dédiée intégrée à GeoNature offrait ensuite la possibilité de rechercher des programmes, d'affiner l'affichage par mots-clés et de sélectionner ceux à importer.

Une fois les programmes choisis, l'utilisateur pouvait définir les filtres à appliquer aux données (la période d'intérêt et les champs souhaités, à l'exception de la *monitoring location*, automatiquement reprise à partir des programmes sélectionnés). Le module interrogeait alors l'api pour récupérer les observations correspondantes.

Seuls les champs nécessaires à l'intégration dans GeoNature étaient retenus, conformément aux exigences du modèle SINP-974. Une transformation était appliquée pour obtenir une structure compatible avec les mécanismes d'import de GeoNature.

Enfin, le module produisait un fichier CSV intermédiaire, destiné à être importé via l'infrastructure existante de GeoNature. Ce fichier constitue un support de contrôle et de préparation, distinct des archives produites lors de l'extraction des données proprement dite.

Chaque opération d'import était consignée dans un historique affiché dans un second onglet, permettant de suivre les actions réalisées, leur date, leur statut et les éventuelles erreurs rencontrées. L'ensemble du module était réservé aux administrateurs, conformément aux pratiques habituelles de contrôle des imports dans GeoNature.

## 4.3. Conception technique du module d'import Quadrige

Le module Quadrige développé durant le stage a été conçu comme un module externe à GeoNature, respectant l'architecture recommandée par le projet tout en prenant en compte les contraintes spécifiques de l'API GraphQL mise à disposition par l'Ifremer. L'objectif principal était de proposer une chaîne d'extraction robuste, traçable et exploitable, sans perturber le cœur applicatif de GeoNature ni les processus existants de gestion des données naturalistes.

#### 4.3.1. Architecture générale et découplage frontend/backend

Le module repose sur une architecture client–serveur classique, intégrée à GeoNature via un *blueprint* Flask côté backend et un module Angular dédié côté frontend. Le backend est responsable de l'ensemble des interactions avec l'API Quadrige, incluant l'authentification, la construction des requêtes GraphQL, le suivi des extractions, la gestion des fichiers produits et l'exposition de routes REST sécurisées.

Le frontend se limite volontairement au pilotage des opérations : sélection des paramètres, lancement des extractions et visualisation des résultats. Il n'accède jamais directement à l'API Quadrige ni aux paramètres sensibles, ce qui permet de renforcer la sécurité globale du dispositif.

Ce découplage permet :

- de centraliser les accès à l'API Quadrige et les jetons d'authentification ;
- de limiter la surface d'exposition des données sensibles ;
- de faciliter la maintenance et les évolutions indépendantes du backend et du frontend.

#### 4.3.2. Gestion de la configuration et des paramètres sensibles

Le module s'appuie sur un fichier de configuration dédié, chargé côté backend via le mécanisme standard de GeoNature (`geonature/config`). Ce fichier centralise l'ensemble des paramètres techniques nécessaires au fonctionnement du module, notamment :

- l'URL de l'API GraphQL Quadrige ;
- le jeton d'authentification requis pour les appels à l'API ;
- des paramètres métiers optionnels, tels que les localisations suggérées ou les champs extractibles.

Une route backend permet d'exposer cette configuration au frontend de manière contrôlée, afin de rendre l'interface dynamique et adaptable. Certaines données métiers restent actuellement définies côté frontend, mais l'architecture mise en place permettrait de les centraliser entièrement côté backend dans une version ultérieure, sans remise en cause du fonctionnement global.

#### 4.3.3. Extraction des programmes Quadrige

La première étape du processus consiste à identifier les programmes Quadrige pertinents pour un territoire donné. Cette extraction repose sur une requête GraphQL spécifique fournie par l'API Quadrige, filtrée à partir d'un préfixe de *monitoring location* correspondant au périmètre géographique ciblé.

L'API Quadrige renvoyant l'ensemble des instances associées à un programme dès lors qu'au moins une station correspond au critère de recherche, un traitement complémentaire est appliqué côté backend. Les fichiers CSV bruts générés par l'API sont filtrés a posteriori à l'aide de

la bibliothèque `pandas`, afin de ne conserver que les programmes effectivement localisés sur le territoire demandé.

Cette étape permet de produire :

- un CSV brut, conservé à des fins de traçabilité et de vérification ;
- un CSV filtré, utilisé pour l'affichage et la sélection des programmes dans le frontend.

Les métadonnées associées à chaque extraction (localisation, horodatage, filtre utilisé) sont sauvegardées afin de permettre la reprise des extractions et la consultation des résultats antérieurs.

La Figure 4.1 illustre le cycle complet d'extraction des programmes Quadrige, depuis la sélection d'un périmètre géographique par l'administrateur jusqu'à la mise à disposition des fichiers CSV exploitables dans GeoNature. Elle met en évidence le fonctionnement asynchrone de l'API Quadrige et le rôle du backend GeoNature dans la transformation et le filtrage des résultats retournés.

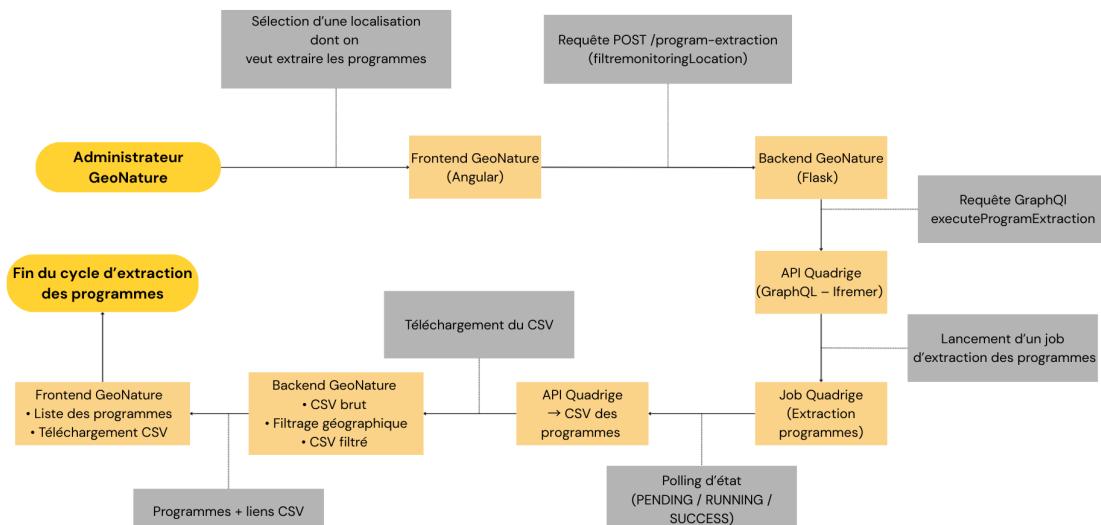


Figure 4.1: Schéma fonctionnel des échanges de données entre GeoNature et le système Quadrige lors de l'extraction des programmes

L'intérêt principal de ce schéma réside dans la manière dont sont traités les résultats retournés par l'API Quadrige lors de l'extraction des programmes. En effet, la requête GraphQL adressée à l'API permet d'identifier les programmes possédant au moins une instance associée à la *monitoring location* sélectionnée par l'utilisateur. Toutefois, la réponse fournie par Quadrige prend la forme d'un fichier CSV contenant l'ensemble des instances de ces programmes, y compris celles localisées en dehors du périmètre géographique ciblé.

Un traitement complémentaire est donc nécessaire côté backend GeoNature. Le CSV brut généré par l'API est d'abord conservé, puis filtré a posteriori afin de ne retenir que les instances

effectivement rattachées à la *monitoring location* sélectionnée. Ce filtrage repose notamment sur l'analyse des codes de localisation, permettant d'exclure les stations hors territoire d'intérêt.

À l'issue de ce traitement, un CSV filtré est produit. Il ne conserve que les informations pertinentes pour la suite du processus, telles que le code du programme, son état, les dates associées, l'organisme responsable et les métadonnées nécessaires à la sélection des programmes dans l'interface GeoNature. Ce fichier constitue la base fonctionnelle utilisée par le frontend pour l'affichage, la recherche et la sélection des programmes à extraire.

#### 4.3.4. Extraction des données et orchestration des traitements asynchrones

L'extraction des données repose sur un mécanisme asynchrone propre à l'API Quadrige. Pour chaque programme sélectionné, une requête GraphQL est soumise afin de lancer un job d'extraction côté serveur Ifremer. Le module implémente un mécanisme de *polling* permettant de suivre l'évolution de l'état de chaque extraction (PENDING, RUNNING, SUCCESS, WARNING, ERROR).

Les extractions sont gérées de manière indépendante, ce qui permet :

- de paralléliser les traitements sur plusieurs programmes ;
- de gérer finement les échecs ou avertissements retournés par l'API ;
- d'éviter un blocage global en cas d'échec partiel.

À l'issue des traitements, les fichiers ZIP générés par l'API Quadrige sont téléchargés et stockés temporairement sur le serveur GeoNature. Chaque fichier est renommé selon une convention explicite intégrant le programme, la localisation et l'horodatage, garantissant ainsi sa traçabilité.

#### 4.3.5. Gestion des états, des erreurs et de la traçabilité

Un soin particulier a été apporté à la gestion des états et des erreurs tout au long du processus d'extraction. Les situations suivantes sont explicitement prises en compte :

- erreurs de communication avec l'API GraphQL ;
- délais excessifs lors des extractions longues ;
- incohérences ou réponses partielles retournées par l'API ;
- extractions aboutissant à des avertissements ou à l'absence de données.

Chaque extraction génère un retour structuré indiquant son statut, les avertissements éventuels et les liens vers les fichiers produits. Ces informations sont transmises au frontend afin de permettre à l'utilisateur d'identifier rapidement les extractions exploitables et celles nécessitant une vérification.

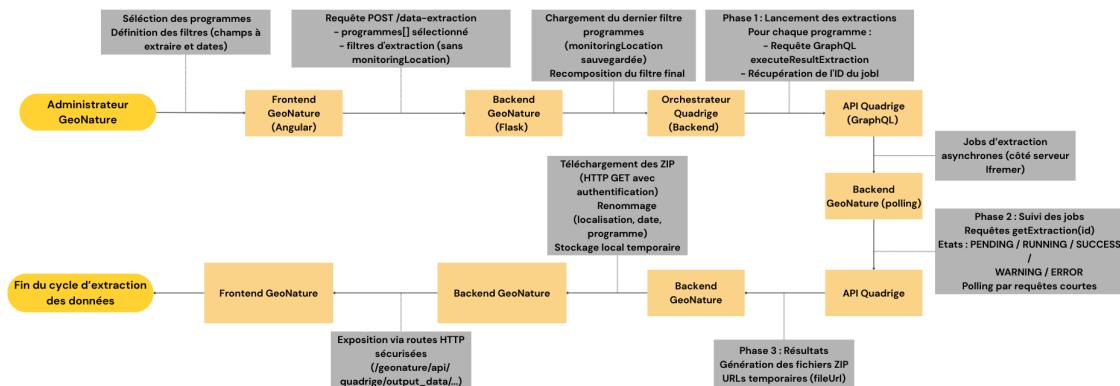


Figure 4.2: Schéma fonctionnel des échanges de données entre GeoNature et le système Quadrigé lors de l'extraction des données des programmes

Ce schéma met en évidence le fonctionnement spécifique de la phase d'extraction des données associées aux programmes Quadrigé.

Lors de cette phase, la *monitoring location* n'est volontairement pas fournie dans les filtres transmis par l'utilisateur. Elle est automatiquement reprise à partir des programmes précédemment extraits et filtrés lors de la phase d'extraction des programmes. Ce choix permet de garantir une cohérence stricte entre le périmètre géographique utilisé pour identifier les programmes et celui appliqué lors de l'extraction des observations.

L'extraction des données est alors exécutée programme par programme, à partir de la liste sélectionnée par l'utilisateur dans l'interface GeoNature. Cette liste provient exclusivement du CSV filtré produit lors de l'extraction des programmes. Seuls les programmes effectivement rattachés au territoire d'intérêt peuvent ainsi être traités.

Ce chaînage explicite entre les deux phases permet d'éviter les incohérences de paramétrage, notamment liées à une erreur de localisation. Il sécurise l'exécution des extractions en s'appuyant sur un périmètre déjà validé, plutôt que sur une saisie manuelle répétée des filtres.

Plus largement, ce schéma met en évidence la séparation volontaire entre la phase d'extraction des programmes et celle de l'extraction des données associées, qui constitue un choix technique structurant du module.

La phase d'extraction des programmes permet d'identifier, filtrer et qualifier les programmes disponibles sur un territoire donné, aboutissant à une liste contrôlée servant de base à l'ensemble des opérations ultérieures.

La phase d'extraction des données s'appuie exclusivement sur cette liste et intervient dans un second temps, une fois les programmes visualisés et sélectionnés. Cette organisation permet à l'utilisateur de conserver une maîtrise fine sur le périmètre fonctionnel des extractions avant de

lancer des traitements potentiellement longs.

#### 4.3.6. Interface utilisateur et pilotage des extractions

Afin de rendre le fonctionnement du module plus concret, les figures suivantes présentent les principales interfaces utilisateur du module Quadrige, depuis l'exploration des programmes jusqu'à l'accès aux résultats d'extraction.

Le frontend Angular du module a été conçu pour accompagner l'utilisateur à travers les différentes étapes du processus : définition des filtres, extraction des programmes, sélection des données et lancement des extractions. Des contrôles de cohérence sont intégrés afin de prévenir les erreurs de paramétrage, notamment sur les champs obligatoires ou les périodes temporelles.

En fonction du périmètre géographique ciblé, l'extraction peut retourner plusieurs dizaines de programmes. Il a donc fallu tenir compte de leur nombre potentiellement élevé, en intégrant des mécanismes de recherche par mots-clés afin de faciliter leur exploration et leur sélection. Il est ainsi possible de filtrer dynamiquement la liste des programmes extraits en saisissant un terme dans la barre de recherche, permettant de cibler rapidement un programme par son nom ou ses métadonnées associées.

Programme	Libellé	Date de début	État	Responsables
CONSTRUI_RUN_PA_POISSONS	Projet CONSTRUI Réunion - Parcours aléatoire poissons	18/06/2024	Actif	DUVAL Magali, MAUREL Laurence, RUNGASSAMY Tewamie
EFFET-RESERVE_LAREUNION_BELT_POISSON	Suivi de l'effet de la Réserve Marine Réunion sur les poissons (BELT)	04/11/2020	Actif	DUVAL Magali, MAUREL Laurence, RUNGASSAMY Tewamie, DUVAL Manali

Figure 4.3: Liste des programmes Quadrige extraits pour un périmètre géographique donné, tri par mots clés avec la barre de recherche

Cette logique de recherche est également appliquée lors des phases de paramétrage. Dans les interfaces de filtrage, l'utilisateur peut rechercher les champs disponibles à l'extraction en saisissant directement du texte dans le champ dédié, ce qui facilite la sélection lorsque la liste des champs est étendue. De la même manière, les localisations suggérées lors de la définition des filtres peuvent être recherchées par saisie textuelle, améliorant l'ergonomie et limitant les erreurs de sélection.

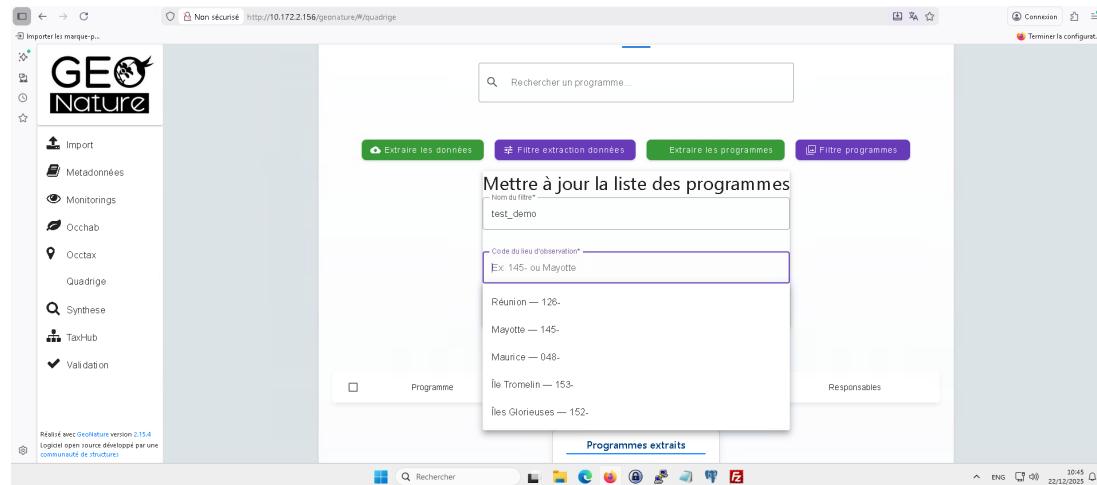


Figure 4.4: Interface de définition des filtres d'extraction des programmes Quadrigé

Les résultats des extractions sont présentés sous forme de liens téléchargeables, accompagnés d'indicateurs visuels signalant les éventuels avertissements. Cette approche permet à l'administrateur de conserver une maîtrise complète sur les données produites avant toute intégration dans GeoNature ou dans des plateformes partenaires telles que Borbonica.

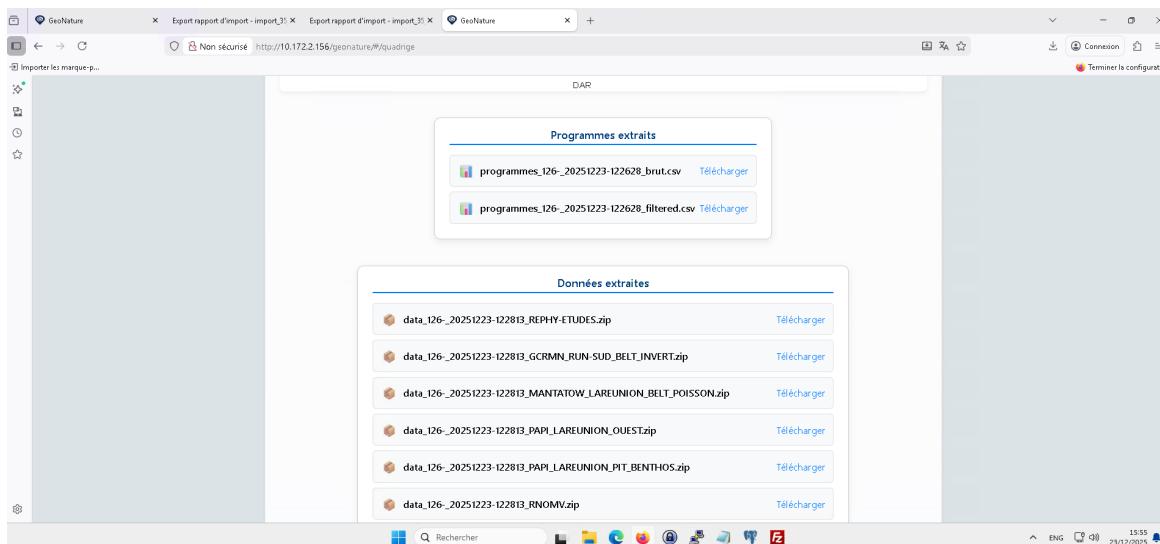


Figure 4.5: Résultats des extractions Quadrigé et accès aux fichiers générés

Une interface spécifique permet également de définir les filtres d'extraction des données associées aux programmes sélectionnés (périodes temporelles, localisations, champs à extraire). À des fins de lisibilité, cette interface est présentée en annexe (voir Figure 7.2).

#### 4.3.7. Synthèse

L'architecture retenue pour le module Quadrige répond aux contraintes spécifiques de l'API Ifremer tout en respectant les principes de modularité, de sécurité et de traçabilité propres à GeoNature. Elle constitue une base technique solide pour des évolutions futures, notamment vers une automatisation partielle des imports, tout en conservant un contrôle humain sur les données produites.

### 4.4. Contraintes, dépendances et livrables du module Quadrige

La réussite du projet dépendait principalement de l'accessibilité de l'API Quadrige, de la stabilité de ses services, et de la disponibilité d'une documentation actualisée. L'intégration dans GeoNature nécessitait également de respecter la structure modulaire du noyau applicatif et les contraintes du modèle SINP-974.

Les livrables attendus comprenaient le code source du module, un fichier de configuration, une documentation à destination des administrateurs (installation, configuration, maintenance) ainsi qu'un guide d'utilisation orienté métier.

Plusieurs évolutions ont été envisagées : automatisation des imports périodiques, gestion des imports incrémentaux, intégration à la gestion des droits de GeoNature et, à plus long terme, publication du module dans le catalogue officiel des extensions GeoNature.

### 4.5. Extension du module api2GN pour l'intégration des données Pl@ntNet

#### 4.5.1. Rôle du module api2GN dans l'écosystème GeoNature

Le module **api2GN** constitue une brique transversale au sein de l'écosystème GeoNature. Il se distingue des modules d'import classiques, généralement fondés sur des fichiers intermédiaires produits manuellement, en permettant l'intégration directe et automatisée de données issues de sources externes.

Ces sources sont exposées via des services web, tels que des API REST (Interface de programmation reposant sur le protocole HTTP, utilisant des opérations standard (GET, POST, PUT, DELETE) pour accéder et manipuler des ressources identifiées par des URL), des flux JSON structurés ou des services WFS. Le module a été conçu pour répondre à des besoins d'automatisation et de récurrence, tout en respectant les contraintes du modèle de données du SINP-974 et de la table *Synthèse* de GeoNature.

Le code source du module **api2GN** est disponible publiquement sur GitHub [Dépôt officiel api2GN](#).

Son fonctionnement repose sur une séparation claire entre deux niveaux. Le premier correspond au cœur du moteur d'import, chargé de l'orchestration des appels aux API, de la gestion des

erreurs, de l'historisation des opérations et de l'insertion des données en base. Le second niveau est constitué de parseurs spécialisés, responsables de la transformation des données sources vers le modèle attendu par GeoNature.

Chaque parseur implémente ainsi une logique métier propre à une source donnée, tout en s'appuyant sur un socle commun. Ce socle prend en charge la gestion des géométries, la résolution des nomenclatures, la validation des mappings, l'historisation des imports et l'intégration aux mécanismes de sécurité de GeoNature.

Dans ce contexte, le module `api2GN` apparaît particulièrement adapté à l'intégration de données issues de plateformes participatives ou ouvertes, telles que Pl@ntNet. Ces plateformes se caractérisent par des volumes importants, une fréquence de mise à jour élevée et des modalités d'accès très différentes de celles des bases institutionnelles comme Quadrige.

#### 4.5.2. Objectifs et positionnement du parser Pl@ntNet

On peut retrouver la documentation du fonctionnement de l'API Pl@ntnet utilisée à cette adresse : [Documentation de l'API Pl@ntNet](#)

Le développement du parser Pl@ntNet s'inscrit dans une logique complémentaire à celle du module Quadrige. Alors que l'import Quadrige vise des données institutionnelles, produites dans un cadre scientifique structuré et contrôlé, les données issues de Pl@ntNet relèvent d'un contexte participatif.

Ce type de données se caractérise par une production massive et continue d'observations, une grande hétérogénéité des contributeurs et une variabilité importante de la qualité des identifications. La fiabilité des données repose en grande partie sur des mécanismes automatisés de reconnaissance et de validation.

Dans ce contexte, l'objectif du parser Pl@ntNet n'était pas de proposer une intégration exhaustive de l'ensemble des observations disponibles. L'enjeu consistait plutôt à mettre en place une chaîne d'import automatisée mais maîtrisée, capable de fonctionner régulièrement, tout en conservant un niveau élevé d'exigence sur la qualité des données intégrées.

Le parser a ainsi été conçu comme un outil configurable et sélectif. Il permet d'appliquer des filtres spatiaux, temporels et taxonomiques, et impose un contrôle strict de la compatibilité des données avec le référentiel TAXREF (Référentiel taxonomique national de la faune, de la flore et de la fonge, maintenu par le MNHN).

Cette approche illustre une stratégie d'intégration différente de celle retenue pour les données Quadrige. Elle repose sur une automatisation plus poussée, compensée par des mécanismes renforcés de validation et de traçabilité.

#### 4.5.3. Architecture technique et principes de conception

Sur le plan technique, le parser Pl@ntNet repose sur l'architecture standard des parseurs du module `api2GN`. Il hérite directement de la classe `JSONParser`, elle-même dérivée de la classe générique `Parser`. Cette hiérarchie permet de mutualiser l'ensemble des fonctionnalités communes aux sources de données exposées sous forme JSON.

Le parser conserve toutefois la responsabilité de la logique métier spécifique à Pl@ntNet. Le déroulement général d'une exécution suit une séquence clairement définie. Après l'initialisation du parser et le chargement de la configuration, les requêtes vers l'API Pl@ntNet sont construites. Les observations sont ensuite récupérées de manière paginée, puis transformées et normalisées. Une phase de résolution taxonomique est alors appliquée, avant l'insertion des observations valides dans la table *Synthèse* de GeoNature.

Un principe fondamental a guidé la conception du parser. Aucune logique métier n'est codée en dur dans le code Python. Les paramètres d'appel à l'API, les filtres appliqués, le mapping des champs et les règles de validation sont intégralement définis dans la configuration.

Ce choix favorise la maintenabilité du module et facilite son adaptation à d'autres contextes territoriaux, sans remise en cause de l'architecture existante.

#### 4.5.4. Configuration dynamique et pilotage par fichier TOML

L'un des apports majeurs du travail réalisé concerne la gestion de la configuration du parser Pl@ntNet. Celui-ci s'appuie sur un fichier `TOML`, chargé via le mécanisme standard de configuration de GeoNature.

Ce fichier permet de définir de manière centralisée les paramètres d'accès à l'API Pl@ntNet, les filtres taxonomiques éventuels, les bornes temporelles d'extraction et l'emprise géographique ciblée, définie sous la forme d'un polygone GeoJSON. Le mapping entre les champs fournis par Pl@ntNet et ceux de la table *Synthèse* est également entièrement décrit dans ce fichier.

Le parser implémente un mécanisme de configuration par défaut.

En l'absence de fichier `api2gn_config.toml`, ou si celui-ci est incomplet, un jeu de valeurs par défaut intégrées au code est utilisé. Ce comportement permet de garantir le fonctionnement du module en environnement de test et de limiter les risques de blocage en production, tout en signalant explicitement les incohérences via les logs.

Une route API dédiée permet de consulter la configuration effective chargée par GeoNature. Cette route applique une validation métier non bloquante, signalant les incohérences potentielles (dates inversées, géométrie invalide, mapping incorrect) sans empêcher l'exécution du parser. Ce choix vise à favoriser la transparence et l'autonomie des administrateurs, tout en conservant un haut niveau de robustesse.

Le parser Pl@ntNet met également en œuvre un mécanisme d'auto-création des métadonnées nécessaires à l'intégration dans GeoNature. Lors de la première exécution, le parser vérifie l'existence de la source, du cadre d'acquisition et du jeu de données associés à Pl@ntNet. En l'absence de ces éléments, ils sont automatiquement créés en base, dans le respect du modèle GeoNature. Ce mécanisme permet de rendre le parser autonome et de limiter les opérations manuelles d'initialisation lors du déploiement.

La structuration des données repose sur le standard Darwin Core, tel que défini par le [GBIF](#).

Cette stratégie renforce la robustesse du module, notamment en phase de test ou de développement, tout en incitant à une configuration explicite en environnement de production. Elle ouvre également la voie à une future interface d'administration graphique dédiée au pilotage de ces paramètres.

La Figure 4.6 synthétise le déroulement général du flux d'import des données Pl@ntNet, depuis l'interrogation de l'API jusqu'à l'insertion des observations validées dans la table *Synthèse* de GeoNature.

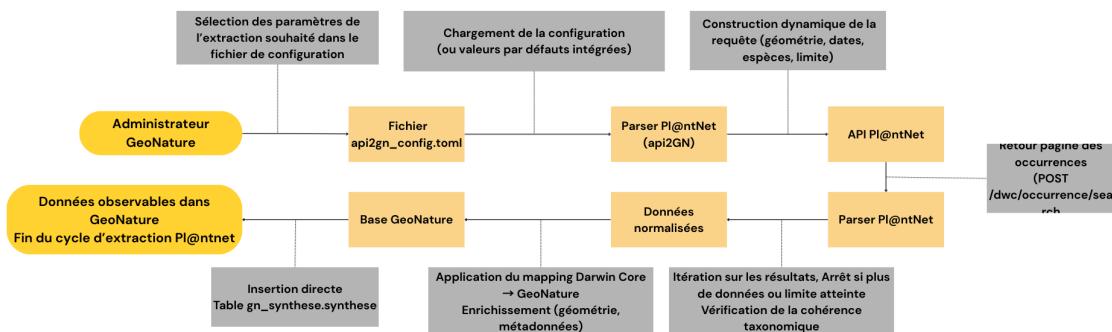


Figure 4.6: Schéma fonctionnel des échanges de données entre Géonature et Pl@ntnet

Ce schéma illustre le fonctionnement global du parser Pl@ntNet et met en évidence le rôle central joué par la configuration dans le pilotage de l'import.

À la différence du module Quadrige, l'import Pl@ntNet repose entièrement sur un fichier de configuration externe.

L'administrateur GeoNature définit les paramètres de l'extraction dans un fichier `api2gn_config.toml`, qui doit être placé dans le répertoire `geonature/config` avant le redémarrage de l'application. Ce mécanisme conditionne l'ensemble du comportement du parser : périmètre géographique, période temporelle, liste d'espèces, volume maximal de données et règles de validation taxonomique.

Ce choix d'architecture permet une extraction entièrement personnalisée, reproductible et traçable, au prix d'une intervention plus technique de l'administrateur. Il constitue un compromis assumé, adapté à un fonctionnement automatisé et périodique, et ouvre la voie à une évolution future vers une interface graphique dédiée, évoquée dans le chapitre consacré aux perspectives.

Le schéma met en évidence l'existence d'un mécanisme de limitation et de pagination des résultats.

tats retournés par l'API Pl@ntNet, sans en détailler le fonctionnement interne. Cette abstraction volontaire permet de conserver une lecture fonctionnelle du flux, centrée sur la construction de la requête et la réception des occurrences.

Concrètement, le contrôle du volume de données importées repose sur une pagination basée sur un couple `limit/offset`, imposé par l'API Pl@ntNet et exploité par le parser. La valeur de la limite est définie par le paramètre `plantnet_max_data` du fichier de configuration. Elle correspond au nombre maximal d'occurrences récupérées lors d'un appel à l'API et conditionne la taille des blocs traités successivement.

Après chaque appel, le parser incrémente automatiquement l'offset afin de solliciter la page suivante de résultats. L'itération se poursuit jusqu'à ce que l'API retourne un nombre d'occurrences inférieur à la limite demandée, ce qui marque la fin des données disponibles pour le périmètre et les filtres définis.

Ce mécanisme permet de maîtriser précisément le volume total d'observations traitées, tout en limitant la charge réseau et mémoire. Un seuil de sécurité supplémentaire est implémenté côté parser afin d'interrompre l'extraction si le nombre de pages devient excessif, prévenant ainsi toute dérive liée à une configuration inadaptée ou à un comportement inattendu de l'API distante.

#### 4.5.5. Résolution taxonomique et contrôle de la qualité des données

La résolution taxonomique constitue l'enjeu central de l'intégration des données Pl@ntNet dans GeoNature. Afin de garantir la cohérence avec le référentiel national TAXREF et les exigences du SINP-974, le parser met en œuvre une chaîne de validation stricte.

Pour chaque observation, le nom scientifique fourni par l'API Pl@ntNet est d'abord normalisé. Les mentions infra-spécifiques et les annotations non standard sont supprimées afin d'améliorer les chances de correspondance avec le référentiel.

Le parser tente ensuite de résoudre le `cd_nom` en interrogeant en priorité le référentiel TAXREF local. En cas d'échec, le service TAXREF-LD (Service web du MNHN exposant le référentiel taxonomique TAXREF sous forme de données liées (Linked Data), permettant l'interrogation et la résolution taxonomique via une API) du MNHN (Muséum national d'Histoire naturelle) est sollicité. Le `cd_nom` obtenu est enfin vérifié dans la base locale de GeoNature.

Afin d'optimiser les performances lors des imports volumineux, un cache mémoire des résolutions taxonomiques est utilisé. Ce cache permet d'éviter des requêtes répétées vers la base TAXREF locale ou les services TAXREF-LD pour un même nom scientifique, réduisant significativement le temps de traitement.

Par défaut, le parser fonctionne en mode strict. Toute observation pour laquelle aucun `cd_nom` valide ne peut être résolu est rejetée. Ces rejets sont systématiquement journalisés, ce qui permet d'identifier les taxons problématiques et d'évaluer la qualité globale des données importées.

La Figure 4.7 détaille la procédure de résolution et de validation du nom scientifique appliquée à chaque observation Pl@ntNet.

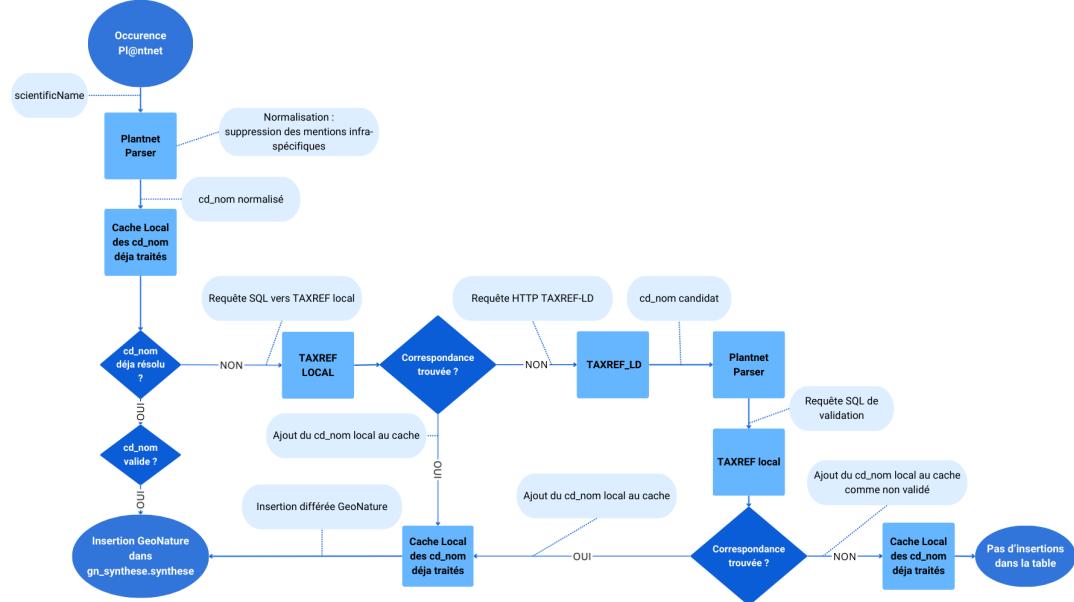


Figure 4.7: Schéma de la procédure de validation du nom scientifique retourné par PI@ntnet

Cette chaîne de validation illustre le rôle clé du référentiel TAXREF dans la sécurisation des imports. Le recours prioritaire au référentiel local, complété par l'appel au service TAXREF-LD, permet de concilier performance et exhaustivité, tout en garantissant la cohérence avec le modèle du SINP-974.

#### 4.5.6. Traçabilité, sécurité et automatisation des imports

Le parser PI@ntNet s'intègre pleinement aux mécanismes de traçabilité fournis par le module api2GN. Chaque exécution est historisée dans une table dédiée. Cette historisation permet de conserver la date du dernier import, le nombre d'observations intégrées et le volume total de données importées.

Un mode *dry-run* est systématiquement disponible. Il permet de simuler un import sans insertion en base. Ce mode constitue un outil essentiel pour tester une nouvelle configuration, évaluer l'impact d'un changement de périmètre ou analyser la qualité des données avant une intégration effective.

Le module intègre un mécanisme d'automatisation basé sur Celery. Les parseurs peuvent être configurés avec une fréquence d'exécution stockée en base. Un processus périodique vérifie les dates de dernier import et déclenche automatiquement les imports lorsque la fréquence définie est atteinte. Il devient ainsi possible d'envisager des imports périodiques automatisés, tout en conservant un contrôle fin sur leur déclenchement et leur suivi.

#### 4.5.7. Apports du travail réalisé

Le développement du parser Pl@ntNet a permis de démontrer la capacité de GeoNature à intégrer des données participatives de manière automatisée, tout en respectant les exigences du SINP-974 en matière de qualité, de traçabilité et de cohérence taxonomique.

Au-delà du cas spécifique de Pl@ntNet, ce travail met en évidence la pertinence de l'architecture api2GN pour l'intégration de sources de données hétérogènes. Il souligne également l'intérêt d'une configuration externalisée pour favoriser la réutilisation du module et son adaptation à différents contextes.

Le parser Pl@ntNet constitue ainsi un prototype fonctionnel et robuste. Il ouvre la voie à l'intégration d'autres sources de données ouvertes et contribue à l'enrichissement progressif de la base GeoNature, notamment pour les taxons terrestres peu couverts par les réseaux institutionnels traditionnels.

Le code source du parser Pl@ntNet et des modules développés durant le stage est versionné et documenté, afin d'en faciliter la reprise et l'évolution, les références étant fournies en Webographie.

## 5. Travaux réalisés et compétences acquises

### 5.1. Synthèse des travaux réalisés

Les travaux réalisés au cours du stage se sont articulés autour de deux axes principaux.

Le premier axe concerne le développement d'un module externe pour GeoNature, dédié à l'exploration et à l'extraction de données marines issues du système d'information Quadrige.

Le second axe porte sur un travail complémentaire autour de l'intégration de données participatives produites par l'application Pl@ntNet. L'objectif était de poser les bases techniques d'un futur parseur, adapté à une intégration automatisée dans GeoNature.

L'ensemble des développements a été mené dans une logique d'intégration au système d'information existant. Les choix effectués prennent en compte les contraintes techniques, organisationnelles et métier propres au SINP-974.

#### 5.1.1. Vue d'ensemble des développements

Les développements réalisés comprennent plusieurs volets complémentaires.

Ils incluent la conception d'un module GeoNature externe, interfacé avec l'API de Quadrige, permettant d'explorer les programmes disponibles et d'en extraire les données associées.

Ils comprennent également la mise en place d'une interface frontend intégrée à GeoNature. Cette interface permet de rechercher, filtrer et sélectionner des programmes et des données à extraire.

Les extractions produisent des fichiers intermédiaires documentés, destinés à une intégration différée dans GeoNature. Cette approche permet de conserver un contrôle sur les données produites avant toute insertion en base.

Enfin, une documentation technique et utilisateur a été rédigée tout au long du projet. Elle décrit les principes de fonctionnement du module, les procédures d'installation, ainsi que les modalités d'utilisation et de maintenance.

L'ensemble de ces travaux a été conçu pour produire un outil fonctionnel, maintenable et réutilisable. Le module est destiné à être repris ou étendu par les équipes de la DEAL ou par d'autres structures partenaires.

### 5.1.2. Travaux principaux : module Quadrige

Les travaux principaux du stage ont porté sur le développement du module Quadrige pour GeoNature.

Ce module permet d'interroger l'API GraphQL mise à disposition par l'Ifremer afin d'identifier les programmes pertinents pour un périmètre géographique donné. Il offre ensuite la possibilité d'extraire les données associées selon des critères définis par l'utilisateur.

Le module repose sur une architecture découpée, séparant clairement le backend et le frontend. Le backend est responsable des échanges avec l'API, de l'authentification, du suivi des extractions et de la gestion des fichiers produits. Le frontend se limite au pilotage des opérations et à la visualisation des résultats.

Les extractions produisent des fichiers intermédiaires, notamment des fichiers CSV et des archives compressées. Ces fichiers sont accompagnés de métadonnées décrivant précisément les paramètres d'extraction utilisés. Cette organisation garantit la traçabilité des opérations et facilite l'analyse des résultats.

Le module a été conçu comme une brique préparatoire. Il ne réalise pas directement l'import des données en base GeoNature, mais vise à sécuriser les étapes amont et à faciliter une intégration future dans le respect des règles du SINP-974.

### 5.1.3. Travaux complémentaires : Pl@ntNet

En parallèle du développement du module Quadrige, un travail complémentaire a été mené autour de l'intégration des données issues de Pl@ntNet.

L'objectif de ce travail était d'évaluer la faisabilité technique d'une intégration automatisée de données participatives dans GeoNature. Il s'agissait également de poser les bases d'un futur parseur utilisable par le module api2GN.

Un client Python générique a été développé pour interroger l'API Pl@ntNet, récupérer les observations et enregistrer les réponses brutes au format JSON.

Ces données ont ensuite été transformées en fichiers CSV conformes au standard Darwin Core. Une attention particulière a été portée à la normalisation des champs, à la gestion des dates, aux coordonnées géographiques et à la génération d'identifiants stables.

Enfin, un pipeline automatisé, intégrant des mécanismes de validation et de contrôle reposant sur une configuration externalisée. Ce pipeline constitue aujourd'hui la base technique du parseur Pl@ntNet intégré au module api2GN.

Bien que complémentaire au module Quadrige, ce travail s'inscrit dans la même logique globale. Il vise à démontrer la capacité de GeoNature à intégrer des sources de données hétérogènes, tout en respectant les exigences de qualité, de traçabilité et de cohérence du SINP-974.

## 5.2. Positionnement et choix d'ingénierie

Le travail réalisé au cours de ce stage ne se limite pas au développement ponctuel d'un script ou à l'utilisation isolée d'une API externe.

Le module Quadrige développé s'inscrit pleinement dans l'écosystème GeoNature. Il respecte l'architecture logicielle existante, les conventions de développement et les mécanismes de sécurité du système.

L'objectif n'était pas uniquement de rendre possible l'interrogation de l'API Quadrige. Il s'agissait de concevoir une chaîne d'acquisition structurée, reproductible et traçable.

Cette chaîne permet d'explorer les données disponibles, de les extraire selon des critères maîtrisés, puis de les préparer pour une intégration ultérieure. Elle laisse volontairement une place au contrôle humain et à l'analyse métier.

Le module a été conçu comme une extension à part entière du système d'information. Il respecte la séparation des responsabilités entre le frontend et le backend, ainsi que les principes de configuration centralisée propres à GeoNature.

Une attention particulière a été portée à la maintenabilité du code. Les choix d'architecture, la structuration des modules et la documentation associée ont été pensés pour faciliter la reprise du projet par d'autres développeurs.

Le travail mené relève ainsi d'une démarche d'ingénierie logicielle. Il vise à produire un outil durable, adaptable à d'autres territoires et cohérent avec les pratiques de gestion des données naturalistes portées par la DEAL et ses partenaires.

## 5.3. Arbitrages techniques

Plusieurs arbitrages techniques ont été réalisés tout au long du stage. Ces choix ont été guidés par les contraintes métier, les usages existants et les exigences de fiabilité associées aux données environnementales.

Un premier choix structurant a consisté à adopter une approche intermédiaire. Le processus repose sur une succession d'étapes distinctes : extraction des données, contrôle, puis intégration différée.

Une automatisation complète et immédiate de l'import en base GeoNature n'a pas été retenue. Ce choix permet de conserver un contrôle humain sur les données extraites, notamment dans un contexte où certaines informations peuvent être soumises à des règles de diffusion spécifiques.

Dans cette logique, l'intégration directe des données Quadrige dans la base GeoNature n'a volontairement pas été implémentée. Le module produit des fichiers intermédiaires documentés, afin de sécuriser les étapes amont et de faciliter l'analyse des résultats.

Concernant le filtrage géographique, un traitement a posteriori à l'aide de la bibliothèque *pandas* a été privilégié. Ce choix s'explique par le fonctionnement même de l'API Quadrige, qui raisonne à l'échelle des programmes et non des stations individuelles.

Enfin, la configuration des modules Quadrige et Pl@ntNet a été entièrement externalisée dans

des fichiers dédiés. Ce choix permet d'adapter les paramètres d'extraction sans modifier le code, de limiter les risques d'erreur et de faciliter le déploiement sur d'autres environnements.

Ces arbitrages traduisent une volonté claire de privilégier la robustesse, la traçabilité et la réutilisabilité, plutôt qu'une automatisation maximale au détriment du contrôle des données.

## 5.4. Difficultés rencontrées et solutions apportées

Tout au long du stage, ces arbitrages ont été confrontés aux contraintes techniques concrètes du projet. Son déroulement a ainsi été marqué par plusieurs difficultés ayant influencé son organisation.

L'absence d'une installation locale stable de GeoNature a constitué l'une des principales difficultés. Les incompatibilités entre certaines versions logicielles, les contraintes liées à Docker et les erreurs du serveur applicatif ont rendu nécessaire la mise en place d'un environnement de travail spécifique.

Cette situation a été renforcée par les contraintes réseau de la DEAL. La présence d'un proxy strict a perturbé l'utilisation de plusieurs outils de développement. La mise en place progressive d'un mécanisme de gestion dynamique du proxy a permis de stabiliser l'environnement.

La migration des serveurs du SINP-974 vers un nouvel hébergeur a également limité l'accès à un environnement de test réaliste. Pendant plusieurs semaines, le développement a dû être poursuivi sans possibilité de déploiement direct.

Une fois l'accès ouvert via une machine virtuelle bastion, les premières manipulations ont été freinées par les restrictions liées à l'accès distant. L'utilisation de Git comme canal principal de transfert s'est révélée déterminante pour poursuivre les tests et ajuster le module.

Une difficulté est apparue en fin de stage, lors du déploiement du module Quadrige sur le serveur de la DEAL.

Le volume important de programmes Quadrige sur le périmètre de La Réunion était déjà connu en amont. Les tests réalisés en local permettaient d'extraire plusieurs dizaines de programmes sans difficulté particulière, y compris lors de traitements longs. Le module avait donc été conçu dès le départ pour gérer un nombre élevé de programmes, supérieur à 80.

Les problèmes sont apparus lors des premiers déploiements en environnement réel. Le serveur GeoNature, basé sur une distribution Debian, utilise une chaîne applicative reposant sur Apache et Gunicorn. Dans cette configuration, des timeouts bloquants sont appliqués aux requêtes HTTP dépassant une durée d'environ 30 secondes.

Lors du lancement des extractions Quadrige en conditions réelles, certaines requêtes longues dépassaient ce seuil. Cela entraînait l'interruption complète du traitement, même lorsque seules quelques extractions restaient en cours. Ces échecs ne reflétaient pas un problème fonctionnel de l'API Quadrige, mais une incompatibilité entre la durée des traitements et les contraintes du serveur applicatif.

Pour répondre à cette contrainte, j'ai revu la logique d'orchestration des appels à l'API Quadrige. Les extractions sont désormais lancées en batch, en soumettant l'ensemble des programmes sélectionnés lors d'une première phase rapide. Un mécanisme de polling global permet

ensuite de suivre l'état de chaque extraction de manière non bloquante, à l'aide de requêtes courtes compatibles avec les timeouts imposés par Apache et Gunicorn.

Cette approche permet de découpler le lancement des extractions de leur suivi, en évitant une logique strictement séquentielle. Les appels lourds à l'API Quadrige sont remplacés par une succession de requêtes courtes de suivi d'état, compatibles avec les contraintes de timeout imposées par Apache et Gunicorn.

Bien que le traitement reste piloté par une requête unique côté GeoNature, cette organisation réduit fortement les risques de blocage liés aux traitements longs. Elle permet également de maîtriser le temps global d'exécution et d'éviter qu'un programme lent n'entraîne l'échec de l'ensemble de l'extraction

Cette solution ne repose pas sur une exécution asynchrone complète côté serveur GeoNature, mais sur une orchestration optimisée des appels à l'API Quadrige. Elle permet néanmoins de contourner efficacement les limitations liées aux timeouts applicatifs en production.

Enfin, la récupération des résultats et le téléchargement des fichiers sont réalisés programme par programme. Les erreurs sont gérées individuellement, ce qui permet de conserver les extractions valides même en cas d'échec partiel.

Cette évolution a permis de rendre le module compatible avec les contraintes de l'environnement de production. Elle a supprimé les timeouts rencontrés lors des premiers déploiements et renforcé la robustesse du module face aux montées en charge, tout en conservant une traçabilité fine des opérations réalisées.

Le travail sur les données Pl@ntNet a également soulevé des difficultés spécifiques. L'API renvoie des structures JSON très hétérogènes, variables selon les taxons et les métadonnées disponibles. Cela a nécessité la conception d'un parseur robuste et flexible.

Par ailleurs, le standard Darwin Core impose un modèle strict, qui ne correspond pas directement aux champs fournis par Pl@ntNet. Un travail de mapping configurable a été nécessaire pour garantir la compatibilité avec GeoNature et Borbonica.

Enfin, l'intégration du parseur au module api2GN a mis en évidence certaines limites dans les migrations SQL existantes. Des ajustements ont été réalisés afin de permettre une installation correcte du module.

Ces difficultés ont nécessité une forte capacité d'adaptation. Elles ont conduit à renforcer la documentation, à multiplier les tests exploratoires et à clarifier progressivement les environnements de travail.

Elles ont également contribué à structurer une démarche rigoureuse, essentielle dans un contexte d'ingénierie logicielle appliquée aux données environnementales.

## 5.5. Compétences acquises et prise de recul

Ce stage m'a permis d'acquérir et de consolider un ensemble de compétences techniques, méthodologiques et transversales, directement mobilisables dans un contexte professionnel d'ingénierie.

Sur le plan technique, j'ai développé une compréhension approfondie de l'architecture de Geo-Nature. J'ai travaillé à la fois sur le backend (Python, Flask, SQLAlchemy) et sur le frontend (Angular, TypeScript), en respectant une séparation stricte des responsabilités.

J'ai acquis une expérience concrète dans l'interfaçage de systèmes hétérogènes, à travers l'utilisation d'API REST et GraphQL. Ce travail m'a permis de mieux comprendre les enjeux liés à l'authentification, à la gestion des erreurs et à la robustesse des échanges réseau.

La manipulation de données volumineuses et hétérogènes m'a conduit à mettre en œuvre des traitements de normalisation, de filtrage et de validation. J'ai notamment travaillé sur des formats standards tels que CSV, JSON et Darwin Core, en tenant compte des contraintes imposées par les référentiels nationaux.

Ce stage m'a également permis de renforcer mes compétences en ingénierie logicielle. La structuration du code, la gestion de la configuration, l'écriture de tests et la documentation ont été des éléments centraux du travail. J'ai appris à concevoir un outil pensé pour être maintenu, repris et adapté dans le temps.

Sur le plan méthodologique, j'ai travaillé en grande autonomie, tout en échangeant régulièrement avec mon tuteur et les agents concernés. J'ai appris à organiser mon travail dans un contexte de projet réel, soumis à des contraintes de délais, d'accès aux environnements et d'incertitudes techniques.

Le stage m'a également permis de mieux appréhender le fonctionnement d'une administration publique. J'ai découvert un cadre de travail où les outils numériques s'inscrivent dans des processus longs, structurés et fortement contraints. Cette expérience m'a aidé à adapter mes choix techniques aux réalités organisationnelles et aux usages métiers.

Au-delà des compétences techniques, ce stage m'a conduit à prendre du recul sur la place du numérique dans les politiques publiques environnementales. Les outils développés ne constituent pas une finalité en soi. Ils participent à la structuration, à la fiabilité et à la diffusion de données qui alimentent des décisions ayant un impact direct sur les territoires et les écosystèmes.

Travailler sur des données naturalistes implique une responsabilité particulière. Les choix techniques effectués influencent la qualité des analyses produites, la visibilité de certaines thématiques et la capacité des acteurs à agir de manière éclairée.

Ce stage m'a ainsi permis de mieux distinguer le fait de produire du code et celui de concevoir un outil réellement utile. Un outil utile est compréhensible, maintenable, adapté à son contexte d'usage et aligné avec les enjeux qu'il vise à servir.

Cette expérience correspond pleinement aux objectifs de la formation d'ingénieur généraliste dispensée à l'ENIB. Elle a renforcé ma capacité à articuler compétences techniques, compréhension des enjeux sociétaux et sens donné aux projets réalisés.

Les compétences acquises et les outils développés au cours de ce stage ouvrent naturellement

---

la voie à plusieurs perspectives d'évolution, présentées dans le chapitre suivant.

## 6. Perspectives

Les travaux réalisés au cours de ce stage ont permis de poser des bases techniques solides pour l'intégration de données externes au sein de GeoNature, en particulier pour les données issues du milieu marin via Quadrige, ainsi que pour les données participatives produites par Pl@ntNet. Plusieurs perspectives d'évolution ont été identifiées, tant sur le plan fonctionnel que technique. Elles visent à renforcer l'ergonomie des outils développés, à améliorer leur intégration dans l'écosystème GeoNature et à faciliter leur appropriation par les administrateurs et gestionnaires de données.

### 6.1. Perspectives d'évolution du module d'import Quadrige

Le module d'import Quadrige développé durant le stage constitue une première brique fonctionnelle permettant d'explorer, d'extraire et de préparer des données marines issues du système d'information Quadrige. Plusieurs axes d'amélioration peuvent être envisagés afin d'en accroître la maturité et de faciliter son intégration opérationnelle à long terme.

#### 6.1.1. Amélioration de l'ergonomie et de l'intégration visuelle

Une première évolution concernerait l'intégration visuelle du module au sein de l'interface GeoNature. L'ajout d'un pictogramme dédié dans la barre de navigation ou dans la liste des modules permettrait d'identifier plus clairement le module Quadrige parmi les autres extensions disponibles.

Ce pictogramme contribuerait à améliorer l'expérience utilisateur en rendant le module plus visible et plus intuitif pour les administrateurs amenés à l'utiliser ponctuellement. Il faciliterait également son appropriation par de nouveaux utilisateurs, notamment dans un contexte de reprise du projet ou de déploiement sur d'autres territoires.

### 6.1.2. Ajout d'une étape de mise en forme et d'intégration des données

À l'heure actuelle, le module produit des archives de données (fichiers CSV bruts, CSV filtrés, fichiers JSON de paramètres) à l'issue de la phase d'extraction depuis l'API Quadrige. Ces archives constituent une étape intermédiaire volontaire, permettant un contrôle manuel des résultats avant toute intégration dans GeoNature.

Une évolution naturelle du module consisterait à ajouter une étape supplémentaire, positionnée après la génération des fichiers ZIP, dédiée à la mise en forme finale et à l'intégration des données dans la base GeoNature. Cette étape pourrait inclure la transformation des données vers le schéma de synthèse, ainsi que leur préparation pour un import direct dans la base de données.

Cette évolution permettrait de proposer une chaîne complète allant de l'extraction à l'intégration, tout en conservant la possibilité de valider manuellement les données en amont. Elle resterait compatible avec les exigences de traçabilité et de contrôle associées au SINP-974.

### 6.1.3. Clarification et extension de la gestion de la configuration

Le module repose déjà sur un fichier de configuration centralisé, chargé côté backend dans l'environnement GeoNature. Ce fichier permet notamment de définir les paramètres techniques essentiels tels que l'URL de l'API Quadrige et le jeton d'authentification, qui sont ensuite mis à disposition du backend.

Le backend expose également le contenu de ce fichier de configuration via une route dédiée, permettant théoriquement au frontend d'accéder dynamiquement à ces informations. Toutefois, par manque de temps, certaines données métier — telles que les localisations suggérées lors du filtrage des programmes ou la liste des champs disponibles pour l'extraction — restent définies directement dans le frontend.

Une évolution souhaitable consisterait à étendre le rôle du fichier de configuration afin d'y inclure ces éléments métier, puis à adapter le frontend pour les charger dynamiquement depuis le backend. Cette approche permettrait de réduire les valeurs codées en dur dans l'interface, d'améliorer la cohérence entre backend et frontend et de faciliter l'adaptation du module à d'autres territoires ou contextes d'usage.

#### 6.1.4. Vers une automatisation partielle et maîtrisée des imports

À plus long terme, le module pourrait évoluer vers une automatisation partielle ou contrôlée des imports, par exemple via des imports périodiques planifiés, des imports incrémentaux basés sur les dates de mise à jour des données, ou des déclenchements manuels avec validation préalable.

Toutefois, certaines spécificités des données Quadrige imposent de conserver un contrôle humain sur les résultats des extractions. Certaines informations peuvent être anonymisées (par exemple les données relatives aux observateurs) ou soumises à des moratoires de diffusion. Il est donc actuellement nécessaire de vérifier les résultats des extractions avant tout traitement visant à leur intégration dans GeoNature.

Ces contraintes devront être pleinement prises en compte dans toute évolution vers une automatisation plus poussée, afin de rester cohérent avec les exigences du SINP-974 et les pratiques de contrôle des données naturalistes.

#### 6.1.5. Renforcement de la stratégie de tests et de validation

Les modules externes GeoNature sont conçus pour s'inscrire dans une logique de maintenance à long terme. À ce titre, la documentation officielle recommande la mise en place de tests automatisés, notamment des tests unitaires et fonctionnels côté backend (via PyTest) ainsi que des tests de bout en bout (*end-to-end*) côté frontend (via Cypress).

Dans le cadre de ce stage, des tests ont été développés et validés sur la version locale du module Quadrige, afin de vérifier le bon fonctionnement des principaux composants : appels à l'API Quadrige, gestion des erreurs, génération des fichiers d'export et cohérence des routes exposées par le backend. De la même manière, des scénarios de tests frontend ont été amorcés pour valider les parcours utilisateurs essentiels.

Toutefois, en raison des contraintes de temps et du calendrier de déploiement sur les serveurs du SINP-974, ces tests n'ont pas pu être intégrés dans la version du module effectivement déployée sur l'environnement serveur.

Une perspective importante consisterait donc à intégrer ces suites de tests dans la version déployée du module, afin de renforcer sa robustesse, de sécuriser les évolutions futures et de faciliter sa reprise par d'autres développeurs. La généralisation des tests automatisés permettrait également de s'inscrire plus étroitement dans les bonnes pratiques recommandées pour le développement et la maintenance des modules externes GeoNature.

Dans l'état actuel, le module reste piloté par une requête HTTP unique côté GeoNature. Une évolution possible consisterait à découpler totalement le traitement de la requête HTTP, par exemple via une exécution en tâche de fond (Celery ou job système), afin de rendre le module pleinement asynchrone côté serveur GeoNature.

## 6.2. Perspectives autour du parser Pl@ntNet et du module api2gn

Les travaux menés sur l'intégration des données Pl@ntNet ont permis de valider la faisabilité technique d'un pipeline complet, depuis l'interrogation de l'API jusqu'à la production de fichiers conformes au standard Darwin Core. Toutefois, ces données étant issues d'une application participative, leur exploitation soulève des enjeux spécifiques de validation et de qualité scientifique.

### 6.2.1. Ajout d'une interface d'administration dédiée

Une perspective majeure consisterait à proposer une interface frontend permettant aux administrateurs de paramétriser le parser Pl@ntNet sans recourir à la modification de fichiers de configuration. Cette interface pourrait permettre de définir les taxons ciblés, les périmètres géographiques, les plages temporelles ainsi que les paramètres de filtrage et de normalisation.

Une telle évolution améliorerait significativement l'ergonomie du dispositif et faciliterait son utilisation par des profils non techniques. Elle permettrait également d'harmoniser l'expérience utilisateur avec celle du module Quadrige, en proposant une logique d'interaction cohérente au sein de GeoNature.

### 6.2.2. Contraintes liées au module api2gn

Dans l'état actuel, cette évolution n'est toutefois pas immédiatement réalisable. Le module **api2gn** est un module officiel de GeoNature, dont l'architecture n'a pas été conçue à l'origine pour intégrer des interfaces frontend spécifiques à chaque source de données.

Toute extension de ce type nécessiterait soit une évolution structurelle du module **api2gn** lui-même, soit la création d'un module externe complémentaire venant piloter le parser Pl@ntNet. Cette réflexion dépasse le cadre du stage, mais les travaux réalisés constituent une base technique solide pour initier ce type d'évolution à l'avenir.

### 6.2.3. Renforcement des mécanismes de validation et d'incrémentalité

Les données Pl@ntNet étant saisies par des utilisateurs non experts, la validation des taxons constitue un enjeu central. Le parser développé intègre déjà un mécanisme de validation basé sur la comparaison des *scientific names* avec les référentiels taxonomiques utilisés par GeoNature, ainsi que sur des interrogations complémentaires de services externes lorsque cela est nécessaire. Un historique des taxons validés et rejetés est également conservé.

Une perspective d'amélioration consisterait à renforcer ces mécanismes, notamment en affinant la gestion des requêtes incrémentales. Actuellement, les extractions sont réalisées par blocs de 1 000 occurrences. Il pourrait être pertinent de permettre une reprise plus fine des extractions interrompues, afin d'éviter de relancer l'ensemble des requêtes en cas d'arrêt partiel du traitement.

### 6.3. Ouvertures transversales

Les développements réalisés durant ce stage ont permis de mettre en évidence l'intérêt d'adapter les stratégies d'intégration des données en fonction de la nature et de l'origine des sources mobilisées. Les travaux menés sur Quadrige et Pl@ntNet illustrent deux approches complémentaires, répondant à des contraintes distinctes.

Pour les données issues de Quadrige, produites dans un cadre institutionnel et scientifique, la priorité a été donnée à la traçabilité et au contrôle des extractions. La séparation entre les phases d'extraction, de préparation et d'intégration permet de conserver un contrôle humain sur des données pouvant être soumises à des règles de diffusion spécifiques (anonymisation, moratoires), tout en préparant leur intégration progressive dans GeoNature.

À l'inverse, les données issues de Pl@ntNet, bien que participatives, peuvent faire l'objet d'un traitement largement automatisé. Le pipeline mis en place intègre des mécanismes de validation taxonomique reposant sur les référentiels utilisés par GeoNature, permettant d'identifier, de corriger ou de rejeter automatiquement les observations non conformes. Les données validées peuvent ainsi être intégrées directement dans la base GeoNature, sans intervention manuelle, tout en conservant un historique des décisions prises par le système.

Cette distinction entre intégration contrôlée et intégration automatisée souligne l'importance d'une architecture d'import flexible, capable de s'adapter aux caractéristiques des sources de données. Elle ouvre la voie à une gestion différenciée des flux, conciliant exigences de qualité scientifique, efficacité opérationnelle et montée en charge des systèmes d'information naturalistes.

## 7. Conclusion

Ce stage s'est inscrit dans un contexte à la fois technique, institutionnel et environnemental exigeant. Il avait pour objectif de contribuer au renforcement des outils numériques utilisés par la DEAL Réunion pour la gestion et la valorisation des données naturalistes.

Les travaux réalisés ont permis de développer une première brique fonctionnelle dédiée à l'exploration et à l'extraction des données marines issues du système d'information Quadrigé. Le module conçu s'intègre à l'architecture de GeoNature, respecte ses principes de sécurité et de modularité, et propose une chaîne d'acquisition structurée, traçable et reproductible. Il constitue une étape préparatoire essentielle à l'enrichissement progressif du SINP-974 en données issues du milieu marin.

En parallèle, le travail mené autour de Pl@ntNet a permis de valider la faisabilité technique d'une intégration automatisée de données participatives dans GeoNature. Le parser développé, adossé au module api2GN, met en œuvre des mécanismes robustes de configuration, de validation taxonomique et de traçabilité. Il illustre une approche complémentaire à celle retenue pour Quadrigé, fondée sur une automatisation maîtrisée et adaptée à la nature des données traitées.

Au-delà des développements réalisés, ce stage a mis en évidence l'importance des choix d'ingénierie dans un système d'information environnemental. La conception des outils, le degré d'automatisation retenu, la gestion des erreurs et la traçabilité des traitements ont un impact direct sur la qualité des données produites et sur les usages qui en découlent. Ces choix doivent donc être pensés en lien étroit avec les contraintes réglementaires, les pratiques métier et les enjeux de diffusion de l'information.

Cette expérience m'a également permis de mieux appréhender le fonctionnement d'une administration publique, où les projets numériques s'inscrivent dans des temporalités longues, des environnements contraints et des logiques de coopération entre acteurs. Elle a renforcé ma capacité à adapter mes solutions techniques à un cadre organisationnel réel, tout en conservant une exigence de rigueur et de qualité.

Enfin, ce stage a contribué à donner du sens à ma formation d'ingénieur généraliste. Il m'a permis de mobiliser des compétences en informatique et en traitement de données au service d'enjeux environnementaux concrets. Il a confirmé mon intérêt pour le développement d'outils numériques utiles, durables et porteurs d'impact, en particulier dans les domaines liés à la gestion des données environnementales et à la préservation des écosystèmes.

Les perspectives identifiées à l'issue de ce travail ouvrent la voie à des évolutions techniques et fonctionnelles ambitieuses, tant pour l'intégration des données marines que pour l'exploitation de sources participatives. Elles constituent autant d'opportunités pour poursuivre l'amélioration des systèmes d'information naturalistes et renforcer leur rôle dans l'aide à la décision publique.

# Annexes

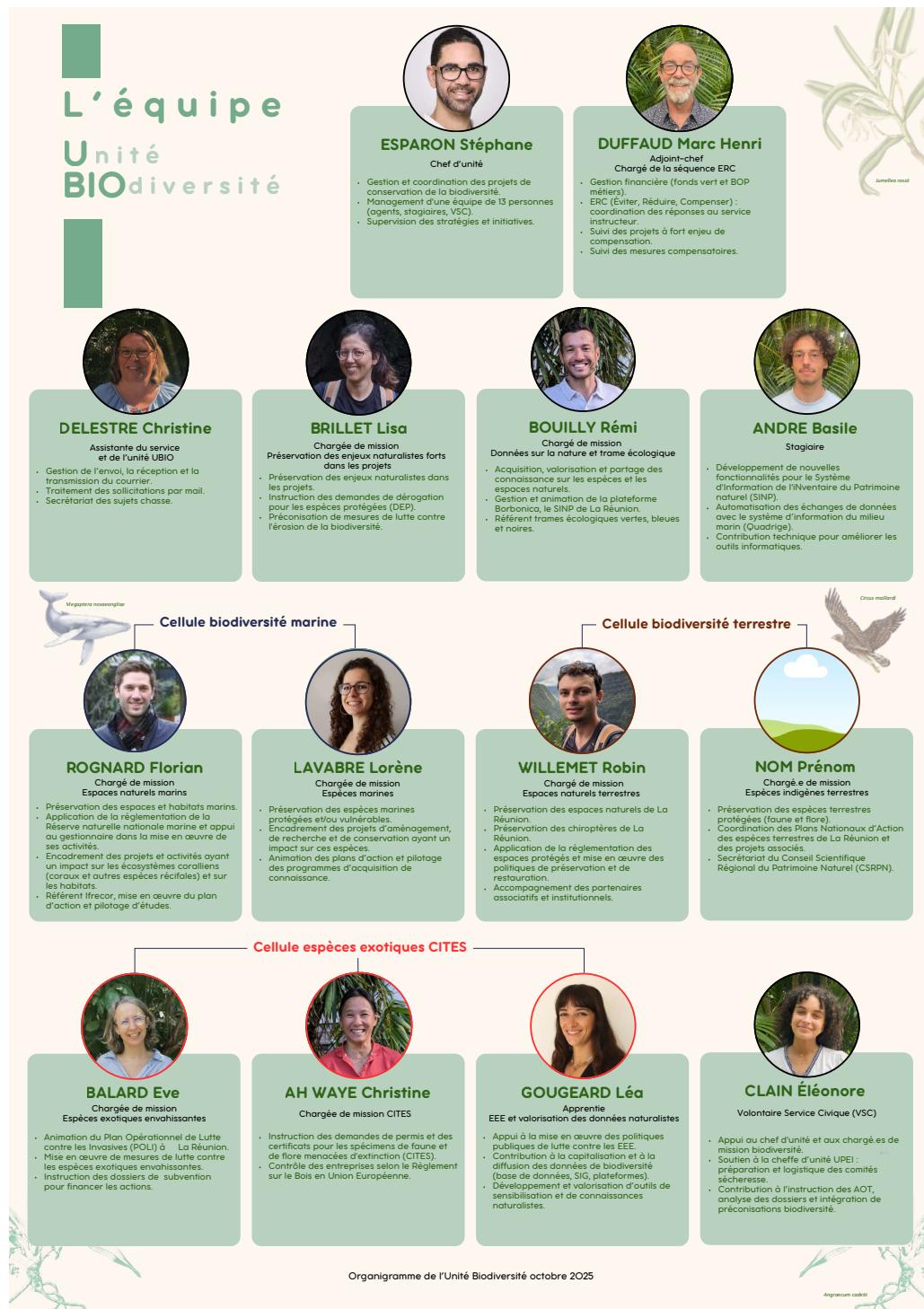


Figure 7.1: Organigramme de l'Unité Biodiversité (UBIO) – SEB / DEAL Réunion

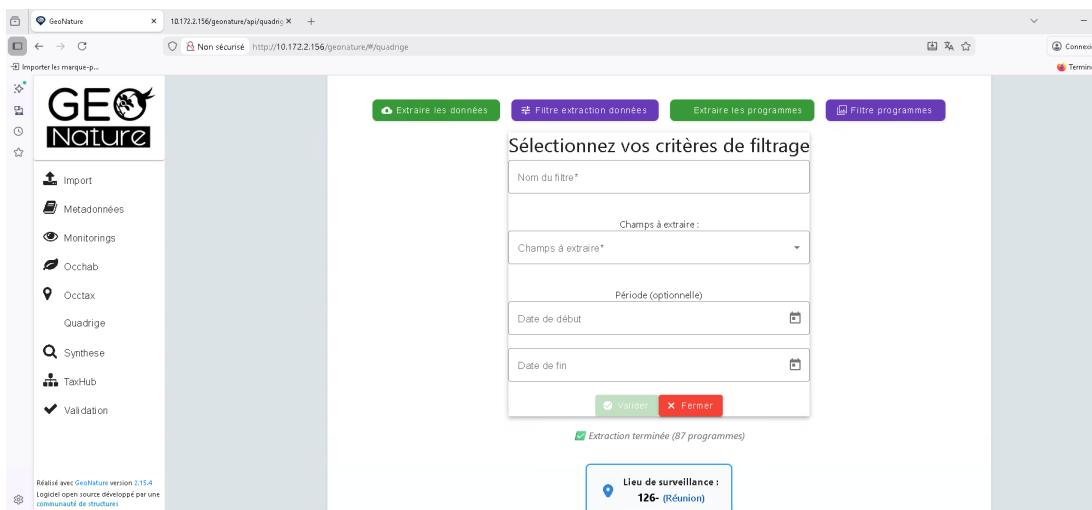


Figure 7.2: Interface de filtrage des données extraites pour les programmes Quadri

## Documentation du module Pl@ntNet

La documentation associée au module Pl@ntNet a été produite sous forme de fichiers Markdown afin de faciliter sa maintenance et sa réutilisation.

Elle comprend :

- une documentation technique décrivant l'architecture du client Python et du parser api2gn
- une documentation utilisateur détaillant les paramètres configurables et les cas d'usage
- un fichier `README.md` présentant le projet, son installation et ses dépendances

L'ensemble de cette documentation est disponible dans le dépôt GitHub, dans le dossier documentation : [documentation](#)

# Webographie

- **Quadrige**

- [Documentation officielle de l'API Quadrige](#)
- [Tutoriels vidéo Ifremer](#)
- [Accès à l'application Quadrige](#)

- **GeoNature**

- [Site officiel GeoNature](#)
- [Dépôt GitHub du projet](#)
- [Documentation générale](#)
- [Guide de développement des modules externes](#)

- **DEAL Réunion**

- [Site officiel de la DEAL Réunion](#)
- [Présentation institutionnelle de la DEAL Réunion](#)

- **Pl@ntnet**

- [Documentation de l'API Pl@ntNet](#)
- [Présentation du format Darwin Core sur le site de GBIF](#)
- [Module api2gn \(officiel GeoNature\)](#)

- **Liens vers les dépôts GitHub des modules développés**

- [Fork personnel du module api2gn \(ajout du parser Pl@ntNet et adaptations des migrations\) Ce module est actuellement intégré dans GeoNature](#)
- **Module Quadrige**
  - \* [Version locale avec tests fonctionnels](#)
  - \* [Version finale intégrée dans GeoNature](#)
- **Module Pl@ntNet**
  - \* [Client Python local, normalisation Darwin Core et documentation](#)