



Parc national des Ecrins

Mise à jour, structuration et diffusion des données du Parc national des Ecrins

Rapport de stage

Raphaël Bres 08/09/2020

Maitre de stage : Camille Monchicourt Tuteur académique : Carmen Gervet

Lettre de mission

Lettre de « mission de stage » - Master 2 Géomatique - 2019/2020

Nom/Prénom étudiant stagiaire : Raphaël Bres

Missions du stage

Objectifs:

Le stage consistera à mettre à jour et structurer différentes données géographiques du Parc national des Ecrins, en vue de mettre en place des bases de données thématiques, consultables et éditables par les agents (PostGIS, QGIS et Lizmap notamment). Il s'agira aussi de renforcer l'ouverture et la diffusion des données du Parc national des Ecrins. Pour cela, un travail sera réalisé sur le catalogage des données, le développement d'outils web de diffusion, ainsi que leur publication sur des plateformes d'opendata.

<u>Tâches confiées au stagiaire – nature de ces tâches :</u>

- Compléter l'inventaire des données géographiques du parc national
- Mettre à jour les données géographiques de référence
- Modéliser et structurer des bases de données par thématique (mesures physiques, agriculture, paysage...)
- Organiser les chaînes de travail de gestion et de mise à jour de ces données

- Déployer et développer des outils de diffusion des bases de données (Lizmap ou applications web Python sur mesure)

- Documenter et illustrer les outils mis en œuvre
- Participer à la modernisation et à la structuration du SI du Parc national des Ecrins

Compétences à mettre en œuvre :

- Maîtrise des logiciels SIG (QGIS notamment)
- Connaissance en conception, modélisation et administration de bases de données
- Gestion de projets
- Structuration et conception d'un système d'information et de chaînes de travail
- Connaissance en SQL (PostgreSQL/PostGIS)
- Connaissance des langages HTML, Python, Javascript, CSS
- Connaissance en ergonomie et architecture web
- Connaissance en administration et sécurité de serveurs web Linux
- Rigueur
- Aptitude à travailler en autonomie et en équipe

Résultats attendus/escomptés en forme de produit :

- Données SIG de référence du Parc national des Ecrins mises à jour (rasters et vecteurs)
- Bases de données thématiques structurées et déployées
- Projets QGIS de référence déployés
- Projets Lizmap en ligne mis en production
- Nouvelles données géomatiques cataloguées et publiées sous licence ouverte

Tuteur de stage dans la structure d'accueil

Nom et Fonction du tuteur :

Camille Monchicourt (Responsable du SI / géomaticien)

Conditions du stage

Lieu et descriptif rapide de la structure d'accueil :

Parc national des Ecrins (Siège) Domaine de Charance 05000 GAP

Équipements géomatiques :

- Poste de travail sous Linux (Ubuntu)
- Tablette
- QGIS
- PostGIS
- PgAdmin
- Python
- Accès à un serveur dédié externe hébergé, disposant de plusieurs machines virtuelles de test et de production pour héberger des bases de données et applications web de géomatique
- Serveur au BRGM avec QGIS server + Lizmap

Dates et durée :

Début : 17 février 2020
Fin : 14 août 2020
<u>Rémunération</u> :
Selon le barème légal en vigueur
Nom de l'enseignant/chercheur chargé du suivi du stage:
<u>Fonction (PR, MCF, chercheur...)</u> : PR Carmen Gervet

<u>Coordonnées</u> : UMR Espace Dev, Maison de la Télédétection 500 rue Jean François Breton 34093 Montpellier Cedex 05

N° Tel, fax et e-mail : Tel 04 67 55 86 51 Carmen.gervet@umontpellier.fr

Il n'y a aucun aspect confidentiel lié à ce rapport.

Remerciements

Je remercie mon maître de stage Camille Monchicourt de m'avoir accueilli au sein du Parc national des Ecrins, pour avoir supervisé mon stage et pour avoir répondu à mes questions lorsque nécessaire.

Je remercie Carmen Gervet pour son encadrement et sa disponibilité durant ce stage.

Je remercie Théo Lechémia pour l'aide qu'il m'a apporté d'un point de vue technique et sa disponibilité malgré le contexte de ce stage.

Je remercie l'ensemble du pole Système d'Informations pour leur chaleureux accueil durant ce stage.

Je remercie Muriel Della-Vedova pour sa disponibilité et ses éclaircissements sur sa demande.

Je remercie Richard Bonnet et Pierre Commenville pour leur disponibilité et leurs éclairages sur la situation de mon stage dans une période mouvementée.

Je remercie Jean François Girres pour m'avoir tenu informé de la situation du master dans la période compliquée que nous avons vécu.

Table des matières

1.		Intro	ntroduction5										
2.		Réali	isatic	n de la demande	7								
	2.1	1.	Cont	texte de l'application	7								
		2.1.1	L.	Architecture et structure de l'application	7								
		2.1.2	2.	Données des constats d'attaques de loups1	1								
		2.1.3	8.	Applications similaires1	3								
	2.2	2.	Réal	isation de l'application	6								
		2.2.1	L.	Base de données	6								
		2.2.2	2.	Développement des fonctionnalités18	8								
		2.2.3	8.	Déploiement sur un serveur	3								
3.		Discu	ussio	n des résultats	5								
4.		Conc	clusic	on21	7								
5.		Bibli	ogra	ohie28	8								
6.		Glos	saire		8								
7.		Anne	exes.		9								
	7.1	1.	Proj	et de modernisation du SIG	9								
	7.2	2.	Com	pte rendu de la réunion de lancement du projet sur l'agriculture et le pastoralisme 3	3								
	7.3	3.	Man	uel utilisateur	4								
8.		Résu	ımé		3								
	8.1	1.	En fr	rançais	3								
	8.2	2.	En a	nglais43	3								

1. Introduction

Le Parc national des Écrins (PNE) se compose du siège se trouvant à Gap, Charance et de maisons du parc se trouvant dans les différentes vallées qui composent son territoire. Le territoire du PNE s'étend sur les vallées de l'Oisans et du Valbonnais pour sa partie Iséroise et du Valgaudemard, du Champsaur, de l'Embrunais, de la Vallouise et du Briançonnais dans le département des Hautes Alpes. Le cœur du Parc national des Ecrins est une zone de très haute montagne avec plus de 150 sommets dépassant 3000 mètres d'altitude et la Barre des Ecrins comme point culminant à 4102 mètres.

Image 1 : Localisation du PNE



Le pôle Systèmes d'Informations (SI) dont je fais partie est rattaché au service scientifique et se trouve au siège du parc national à Charance. Il sert à récolter, structurer, stocker, analyser, suivre et restituer les données du Parc national des Ecrins dans les différentes disciplines et d'assurer la gestion du système d'information, l'informatique et la téléphonie.

L'objectif de ce stage est d'apporter mes compétences afin de moderniser le SIG au sein du parc national, publier des données ainsi que mettre en valeur des données sur des thématiques laissées de côté pour le moment. Ce stage de six mois se découpe en trois projets qui sont :

- La modernisation du SIG du parc et la diffusion des données.
- La structuration et le partage des données sur l'agriculture et le pastoralisme.
- La structuration et le partage des données climatologiques.

La modernisation du SIG consiste d'abord à étudier différents types de fichiers de stockages de données géographiques pour proposer une solution optimale pour une utilisation des agents du Parc national des Ecrins. Il y a aussi la mise à jour des données provenant de l'IGN et la diffusion de données en Open Data.

Ensuite, le projet de structuration et de partage de données sur l'agriculture et le pastoralisme consiste à créer une interface simple de consultation et de récupération des données thématiques pour les agents du parc en priorité. Ce projet a débouché vers un autre besoin plus important qui est une application de suivi des constats d'attaques de loup. Ce projet a rapidement pris le nom « GeoConstats ».

Le projet de structuration et de partage de données climatologiques consiste à créer une application web pour suivre les données transmises par des stations météorologiques se trouvant dans le Parc national des Ecrins.

Avec ces objectifs et en concertation avec mon maître de stage, j'ai élaboré un diagramme de GANTT pour planifier la réalisation de ces trois projets durant les six mois de stage. L'image cidessous présente le calendrier de mon stage.

	_ .		Février Mars			Avril				Mai					Jui	n		Juillet				Août			
	Taches	17	24	2	9	16 2	3 30	6	13	20 27	7	4 11	18	25	1	8	15	22 2	9	6	13	20	27	3	10
										TE	LE	TRA	VAIL												
1	Modernisation et diffusion du SIG																								
1.1	État des lieux du SIG																								
1.2	Migration du shapefile vers le Géopackage																								
1.3	Création de modèles de projets QGIS																								
1.4	Catalogage et ouverture des données																								
1.5	Mise à jour des fonds de carte IGN																								
2	Agriculture																								
2.1	Analyse des besoins																								
2.2	Analyse de l'existant																								
2.3	Base de données																								
2.4	Projet QGIS (seulement valorisation données)																								
2.5	LizMap (seulement valorisation données)																								
2.6	Programmation (seulement constats loups)																								
3	Climatologie																								
3.1	Analyse des besoins globaux																								
3.2	Analyse des besoins spécifiques																								
3.3	Analyse de l'existant																								
3.4	Acquisition traitement des données																								
3.5	Base de données																								
3.6	Programmation																								

Image 2 : Diagramme de GANTT de mon stage à mon arrivée

Ce diagramme présente bien la répartition des trois projets que j'ai à effectuer. Cependant, suite à une réunion avec la commanditaire du projet sur l'agriculture (le compte rendu est en annexe 2 de ce document), nous avons identifié le projet sur le suivi des constats d'attaque de loups comme prioritaire. De plus, ce projet était le plus facilement réalisable en télétravail et cette situation a duré plus de la moitié de mon stage. Ce sont ces éléments qui font que mon stage s'est centré sur le projet de suivi des constats d'attaque de loups et en conséquence, la partie technique de ce document sera centrée sur ce projet. La partie de modernisation des données géographiques sera traitée en annexe de ce document. Le projet sur la climatologie est laissé de côté mais l'application réalisée pour GeoConstats doit être réutilisable pour ce projet.

2. Réalisation de la demande

2.1. Contexte de l'application

2.1.1. Architecture et structure de l'application

Actuellement, le Parc national des Ecrins n'a aucun outil de visualisation et de suivi des constats d'attaques de loups. Seul un fichier au format tableur permet d'obtenir des données analysables. Pour la localisation géographique, c'est encore pire car on ne dispose certaines fois que d'un lieudit qui n'indique aucune appartenance à une commune ou au cœur du parc. Cependant, la saisie d'un constat étant déjà longue, il me faut créer un outil simple qui soit un parfait complément de la méthode traditionnelle de saisie.

L'application doit être simple avec une carte présentant les constats, une liste permettant de visualiser les informations d'un constat, des filtres attributaires pour ne faire afficher que certaines données et des liens pour ajouter, modifier et supprimer un constat. Il faut également proposer un module d'export de données pour les exploiter sur un tableur ou sur un système d'information géographique. Un besoin exprimé est aussi une carte localisant un constat précis avec ses informations dans le but d'imprimer la page pour aller sur le terrain. L'application devra également être liée au système d'identification du Parc national des Ecrins (Users Hub) pour qu'une personne ayant les droits puisse ajouter un constat ou modifier, supprimer un constat. De plus, il faut que le code de cette application puisse être facilement réutilisable pour une application à d'autres données comme celles de climatologie par exemple. Il est aussi important de noter que cette application est développée avec des technologies libres et est publiée sous licence libre pour pouvoir être réutilisée par d'autres structures.

Rentrons maintenant dans la manière dont cette application sera utilisée. Nous allons garder cette application en interne dans un premier temps car elle servira beaucoup pour le personnel du Parc

national des Ecrins uniquement. Pour comprendre les utilisations possibles, voici ci-dessous une image des cas d'utilisations de l'application.



Image 3 : Diagramme de cas d'utilisations de l'application GeoConstats

On a d'abord l'utilisateur lambda peut juste consulter et filtrer les données de constats et les télécharger. Ensuite, on a l'agent recenseur qui peut créer des données ainsi que mettre à jour et supprimer ses données en plus des droits de l'utilisateur de base. Pour finir, il y a l'administrateur qui a l'ensemble des droits de l'agent recenseur mais il peut modifier et supprimer n'importe quel constat.

Concernant l'architecture de l'application, on retrouve ici une architecture client / serveur. Le client émet une requête qui est transmise au serveur (Flask) via le protocole http. Ensuite, le serveur utilise l'ORM (Object Relationnal Mapping) SQLAlchemy pour communiquer avec la base de données (PostgreSQL, PostGIS sur le serveur du BRGM). La base de données retourne un résultat qui est traité par le serveur et le résultat de la requête est envoyé au client sous forme de page web.

Pour le serveur, j'utilise le Framework Flask qui est déjà utilisé par certaines applications développées par le Parc. De plus, ce module présente l'avantage d'être écrit en Python, ce qui est un point fort quand on connait les possibilités offertes par ce langage pour manipuler des données géographiques. On retrouve entre autres des modules pour QGIS et ArcGIS développés en Python, ce qui montre que l'on a un langage adapté au projet. De plus, certains modules facilitent le

traitement de fichiers géographiques comme Shapely ou GeoAlchemy. Le schéma ci-dessous présente le fonctionnement de mon application.

Image 4 : Schéma de l'application GeoConstats



On a ici le schéma d'une application avec une architecture client / serveur classique. Concernant les technologies utilisées pour le projet GeoConstats, on a le côté client avec des pages web développées en HTML avec du JavaScript. La librairie Leaflet permet de placer une carte sur une page web. La librairie Bootstrap est une librairie CSS. Elle facilite la mise en page et la mise en place du style sur une page HTML. Pour le serveur, le module Python Flask est utilisé. Il est présent dans les dernières applications développées par le PNE. Le framework Flask s'appuie sur des librairies existantes. Jinja est un moteur de template qui permet de générer des pages HTML à partir d'un objet envoyé sous forme d'objet Python. On a aussi le package SQLAlchemy qui traduit le modèle relationnel d'une base de données en objet Python. Enfin, pour le système de gestion de base de données, j'utilise PostgreSQL avec sa cartouche spatiale PostGIS. Cet outil me permet de stocker et d'utiliser facilement des géométries dans une base de données, ce qui est requis pour ce projet.

Il est temps d'évoquer les premiers plans de l'application. On retrouve trois types de pages. D'abord, on a les pages de visualisation de l'ensemble des données. Cette page servira de page d'accueil pour accéder à l'ensemble des fonctionnalités de l'applications. L'image ci-dessous présente le format de page prévu.

Image 5 : Page de visualisation de toutes les données

	Connexion	
Filtres Export		
Liste	Carte	

On a sur cette page la carte qui occupe la majeure partie de l'espace avec un formulaire de filtrage et la liste des données. Dans un style plutôt similaire, il y a la page de visualisation pour une seule donnée. Cette page sera utilisée pour la saisie d'un nouveau constat, la modification d'une donnée et pour accéder à des informations détaillées sur un constat. L'image ci-dessous présente cette page.





Comme expliqué précédemment, c'est une page assez similaire à la page de visualisation de l'ensemble des données si ce n'est que le bloc à gauche de la carte n'a plus de formulaire de filtrage mais seulement une seule zone avec le formulaire de saisie ou les informations d'un constat. Le dernier type de page est le tableau de bord. Il est ici pour fournir de manière rapide des données aux agents et aux décideurs. L'image ci-dessous présente la page de tableau de bord.

Image 7 : Schéma de la page de tableau de bord

Barre de navigation	Connexion
Filtres	
Tableaux	

Nous sommes ici sur la page avec le moins d'éléments qui sont juste les tableaux dynamiques et un formulaire de filtrage pour faire évoluer les données des tableaux.

Maintenant que j'ai expliqué l'architecture et l'environnement de l'application, parlons maintenant des données à mettre en valeur.

2.1.2. Données des constats d'attaques de loups

Un constat, comme lors d'un accident de voiture, est un papier demandant beaucoup d'informations sur l'origine de l'attaque ou sur le propriétaire des animaux tués par exemple. Mon application ne reprendra que les données principales. Les constats d'attaques de loups se distinguent en deux catégories : les constats « normaux » et les constats déclaratifs. Les constats normaux sont saisis par des agents du Parc national des Ecrins, de parcs naturels régionaux (PNR) comme celui du Queyras ou de l'Office français de la biodiversité (OFB) qui viennent constater les animaux morts. Normalement, il faut localiser précisément chaque animal mort ou blessé. Cependant, dans le cadre d'un nombre important de décès, l'agent constatant l'attaque place un point pour tous les animaux touchés. Les constats déclaratifs sont des constats avec peu de victimes. Ils peuvent directement être saisis par le propriétaire des animaux visés. Cependant, la localisation est ici beaucoup plus vague car on ne recense qu'un lieu-dit avec une vague description du lieu de l'attaque.

Les données présentées ici sont fournies par la Direction Départementale du Territoire (DDT). Il faut déjà prendre en compte que les données viennent de deux départements donc il y a deux sources de données (la DDT des Hautes Alpes et la DDT de l'Isère). Cependant, les données fournies n'ont aucune dimension géographique et il est donc très compliqué d'obtenir des statistiques par secteur ou par département. On a aussi une grande différence entre la qualité des données fournies par la DDT des Hautes Alpes et la DDT de l'Isère. Du côté isérois, les données sont dans l'application MapLoup alors que du côté des Hautes Alpes, les données ne sont fournies que dans un tableau. L'image ci-dessous présente un exemple des données anonymisées des attaques de loups fournies par la DDT des Hautes Alpes au 17 juillet 2020.



Image 8 : Données anonymisées fournies par la DDT des Hautes Alpes

On constate que l'on ne peut pas facilement obtenir des données par zone géographique. C'est pour cela que je dois réaliser une application permettant une saisie simple venant compléter la fiche de constat envoyée à la DDT. De plus, pour éviter des erreurs de saisie, je vais devoir calculer les champs liés à la géographie comme la commune ou le secteur de l'attaque.

2.1.3. Applications similaires

Afin de bien cibler le résultat attendu, j'ai à ma disposition trois applications qui traitent les mêmes données sur un autre territoire ou qui ont une forme similaire au résultat attendu mais sont utilisées pour d'autres domaines.

Commençons avec l'application Maploup. C'est une application accessible au grand public qui permet de consulter les données des attaques de loups sur les départements de l'Isère, de la Savoie et de la Haute Savoie. Cela inclue donc une partie du territoire du Parc. Ici, nous sommes sur un site très développé avec un grand historique de données. L'image ci-dessous présente la carte interactive des constats liés au loup sur l'année 2020.





On constate ici un dégradé de couleurs en fonction de l'ancienneté de l'attaque et des cercles proportionnels en fonction du nombre de victimes (morts et blessés confondus). On a ici une application très chargée et l'interface grand public ne permet pas la saisie de données donc nous n'avons qu'une partie de la solution envisagée dans cette application. Maintenant, je vais présenter une application sur une autre thématique mais qui se rapproche du résultat attendu.

L'application Géol'Ecrins est développée par le Parc national des Ecrins pour présenter les sites géologiques présents sur le territoire. On retrouve ici une carte créée avec Leaflet et une liste des sites géologiques avec la possibilité en cliquant sur un site, que ce soit sur la carte ou dans la liste, d'accéder à ses informations. Les points forts dans l'utilisation de ce site est l'accessibilité de son

code sur Github (http://github.com/PnEcrins/GeoSites) et son interface proche de ce que je dois développer. L'image ci-dessous présente l'interface de Géol'Ecrins.



Image 10 : Interface de l'application Géol'Ecrins

On a ici une application très basique qui se rapprochera du résultat attendu. Concernant le code de cette application, le côté serveur est réalisé avec Angular.js mais maintenant, le Parc national des Ecrins se dirige vers Flask comme serveur. En lien avec Flask, l'application suivante qui me sert de base de développement utilise cette technologie.

Afin de bien démarrer le projet, j'ai à ma disposition une petite application (https://github.com/PnEcrins/Flask-leaflet-example) pour placer sur une carte des données issues d'une table sous PostgreSQL et remplir un formulaire qui va directement remplir cette table. Ce projet utilise les technologies que je vais mettre en œuvre. Cependant, cette application a un défaut majeur qui est que les points placés sur la carte sont dans la base de données un champ longitude et un champ latitude. On ne retrouve aucune utilisation de PostGIS dans la base de données. L'image ci-dessous présente la base du projet GeoConstats dans l'état dans lequel je l'ai récupéré.



Image 11 : Base du projet GeoConstats, interface de visualisation

On retrouve ici une simple carte avec les données issues d'une table sous PostgreSQL. Le gros point positif de cette application réside dans le formulaire de données qui est opérationnel pour des attributs non géométriques. L'image ci-dessous présente le formulaire de saisie quand j'ai récupéré l'application.

€ ⇒ ⊂ @	0 (localhost:5000/add		🖂 🕁	IN ED 📽 🗏
Microblog: Map Add to map		Add to map		
		Name		
		Email		
		×		
		Y		
		Add to map		

Image 12 : Base du projet GeoConstats, interface de saisie de données

On a un formulaire avec un bouton pour soumettre les données. Le problème de cette page est l'absence de carte pour la saisie de la géométrie des constats. Cette application, même si elle est loin d'être parfaite, me donne une bonne base pour démarrer le projet GeoConstats.

Maintenant que l'on a vu le contexte et l'architecture de l'application GeoConstats, il est temps d'expliquer la manière avec laquelle j'ai réalisé ce projet.

2.2. Réalisation de l'application

2.2.1. Base de données

L'application reposant sur une base de données, il faut d'abord la construire. Même si certaines couches se trouvent déjà sur le serveur que je vais utiliser pour d'autres applications (contour du PNE, limites des secteurs), il me faut créer les tables liées aux constats.

Le Parc national des Ecrins a décidé de créer une sémantique dans la nomination des tables de base de données. Pour une table classique, elle reçoit le préfixe « t_ ». Dans ces tables, on retrouve certains champs ayant plusieurs possibilités de réponse comme le champ « statut » de la table « t_constat » qui peut être « indemnisé », « rejeté » ou « en attente ». Dans ce cas, on crée une table avec le préfixe « bib_ ». Cela permet par exemple de mettre à jour un champ de formulaire avec plusieurs réponses juste en effectuant un changement dans la base de données donc sans toucher au code de l'application. On retrouve aussi le préfixe « l_ » qui signifie ici layer donc qui représente les couches du référentiel géographique. Pour utiliser certaines tables dans le cadre de plusieurs applications hébergées sur le même serveur de base de données, toutes les tables composant le référentiel géographique sont stockées dans un schéma à part. Après avoir expliqué le fonctionnement de l'application et de la base de données, je vais détailler le contenu des tables que j'ai créé à partir des données de constats.

Pour les constats, on retrouve beaucoup de champs en commun entre les constats et les déclaratifs. La table est constituée d'un identifiant qui sert de clé primaire (id integer), d'une date d'attaque (date_attaque date), d'une date de constat (date_constat date), d'un propriétaire (proprietaire string), d'un nombre d'animaux tués (nb_victimes_mort integer), d'un nombre d'animaux blessés (nb_victimes_blesse integer), d'un type d'animaux (type_animaux integer), d'un statut (statut integer) et d'une géométrie (the_geom_point point 2154). Les champs type_animaux et statut sont spéciaux car ils font référence à une table dans laquelle on trouve les possibilités de réponse pour ses champs. Pour la table des constats uniquement, on deux champs réservés aux agents recenseurs (nom_agent1, nom_agent2 string) et un nombre de jours pris par l'agent pour faire le constat (nb_jour_agent float). Pour les constats déclaratifs, on a le lieu-dit de

l'attaque (lieu_dit string). On retrouve aussi des champs qui sont calculés par PostGIS qui sont le secteur du constat (id_secteur integer), la commune du constat (id_commune integer) le département qui est déduit de la commune (departement string), la présence ou non dans le cœur du parc (dans_coeur boolean) et la présence ou non dans l'aire d'adhésion (dans_aa boolean). On a aussi le code de l'utilisateur ayant saisi le constat (id_role integer).

Afin d'automatiser la saisie des champs dépendant de la géométrie (id_secteur, id_commune, departement, dans_coeur, dans_aa) et du nom du créateur du constat dans la base de données (id_role), j'ai créé deux triggers (un pour les constats et un pour les déclaratifs). Ces triggers doivent réaliser des intersections avec les tables du référentiel géographique afin de retrouver le secteur et la commune où un constat a eu lieu. Pour le département, on le récupère depuis le code INSEE de la commune. On teste également la présence dans le cœur et l'aire d'adhésion du PNE (<u>https://github.com/PnEcrins/GeoConstats/blob/master/data/scriptSQL_Constats.sql#L23</u>). Ces triggers ne s'exécutent que lorsqu'un constat est créé ou que le point d'un constat est modifié.

Pour réaliser les procédures requises pour les triggers j'ai dû apprendre le langage PL/pgSQL. C'est un langage de programmation qui permet de réaliser des instructions comme dans un langage classique comme le Java ou le Python mais facilite l'utilisation du SQL. Par exemple, on peut utiliser des conditionnelles afin de vérifier si la valeur d'un champ a été modifié afin d'exécuter ou non une requête SQL.

Le principe est différent pour un ajout de données et pour une mise à jour. Pour l'ajout, on utilise directement la requête SQL de type INSERT INTO alors que pour la mise à jour, on ne va exécuter le trigger que si la géométrie a changé.

Pour calculer les champs géographiques évoqués précédemment, j'ai eu besoins de données géographiques comme les communes ou encore le contour du PNE ainsi que son aire d'adhésion. Pour stocker ses données, certaines applications développées par le Parc national des Ecrins utilisent un schéma nommé ref_geo. Ce schéma dispose d'une table principale nommée l_areas qui stocke toutes les aires géographiques que l'on souhaite. Dans cette table, on a un champ id_type qui est une référence à la table qui stocke les informations sur le type d'aire géographique d'une ligne. Cette table s'appelle bib_area_type.

En plus de cela, il me faut penser à l'inclusion de la gestion des droits d'accès aux différentes fonctionnalités de l'application. C'est pour cela que j'ai importé le modèle de base de données de UsersHub. UsersHub est le système d'identification utilisé sur la majorité des applications du Parc

national des Ecrins. Il est composé d'un schéma de base de données et d'une interface graphique pour faciliter la gestion des droits. Dans le cadre du développement de GeoConstats, je n'utilise que le schéma de base de données mais lors du déploiement, mon application sera connectée à la même base de données d'identification que les autres applications du PNE. Pour installer ce schéma, il existe déjà des scripts SQL à lancer depuis PgAdmin. Cela me créé d'abord les tables avec les contraintes puis cela m'ajoute des données dans ces tables.

La base de données disposant de plusieurs schémas est difficile à mettre en place. C'est pour cela que j'ai créé un script en langage Bash (https://github.com/PnEcrins/GeoConstats/blob/master/install_db.sh) permettant d'installer la base de données juste en exécutant un fichier dans un terminal. Ce script va se servir de scripts SQL existants pour d'autres applications du PNE comme par exemple GeoNature. GeoNature est une application open source permettant le suivi de la faune et de la flore sur un territoire donné. On dispose après l'exécution du script d'une base de données prête à l'emplois dans l'application.

Ensuite, il me faut faire des requêtes sur ma base de données depuis le serveur pour que ces données arrivent sur mon application. Pour cela, le module SQLAlchemy de Python propose une fonction de requête simplifié. Il me faut d'abord créer une nouvelle classe qui va prendre comme attributs les champs de la table qui m'intéresse. Ensuite, le module SQLAlchemy me permet de réaliser des requêtes comme je le souhaite avec par exemple des filtres ou des fonctions à disposition dans PostgreSQL (distinct) et PostGIS (ST_AsGeoJSON, ST_Transform).

Le point plus technique est lors du filtrage des données. A ce moment, il faut d'abord vérifier que les paramètres soient bons puis les ajouter à la requête avant son exécution. Cependant, une requête peut se construire en plusieurs instructions avec SQLAlchemy. Cela veut dire que je peux faire une base pour ma requête sans filtre puis réaliser une instruction conditionnelle et si je rentre dans ma conditionnelle je peux ajouter un filtre. Une fois tous les filtres ajoutés, je n'ai plus qu'à exécuter la requête.

2.2.2. Développement des fonctionnalités

Les fonctionnalités devant être développées sur mon application sont la visualisation, l'ajout, la mise à jour, la suppression, le filtrage et l'export de données avec la possibilité d'imprimer la carte. Cette sous partie va expliquer comment se passe le développement de ces fonctionnalités d'un point de vue global en expliquant les spécificités de certaines routes. Pour la visualisation de données, je dois d'abord faire afficher l'ensemble des données sur la carte. Ensuite, je dois proposer un accès à toutes les fonctionnalités possibles pour une donnée (consultation, modification, suppression). Je dois aussi créer une liste regroupant l'ensemble des données. Il y a aussi la création d'un formulaire de filtrage. Ce formulaire doit, au clic sur un bouton ne faire afficher dans la liste et sur la carte, seulement les données correspondant à l'ensemble des filtres demandés par l'utilisateur. Pour filtrer les données, je réalise une requête de type GET pour envoyer les données au serveur et le laisser ajouter les filtres correspondants dans la requête sur les constats. Il y a de plus l'option de récupérer les données brutes dans un format pour tableur. Une fois le clic sur le bouton d'export effectué, l'application filtre les constats avec les données du formulaire et renvoie un fichier d'extension csv. L'image ci-dessous présente la page de visualisation des constats.



Image 13 : Page de visualisation des constats

On peut voir ici la carte avec le formulaire et la liste des données. Pour mieux visualiser les données sur la carte, j'ai utilisé la librairie Leaflet Marker cluster qui regroupe les points se trouvant dans une zone rapprochée.

On peut aussi ajouter un constat. On retrouve sur la page web la carte avec sur sa gauche le formulaire de saisie. Il est important de voir que le bouton de validation du formulaire n'est pas cliquable tant qu'on n'a pas localisé le constat sur la carte.

Image 14 : Page d'ajout d'un constat



On constate la présence d'indications pour remplir le formulaire de saisie. Ensuite, si l'on a commis une erreur de saisie, on peut encore la modifier depuis la page de visualisation d'un constat en cliquant sur le bouton « Modifier le constat » se trouvant à côté de la donnée avec une erreur. Cela nous amène sur une page similaire à celle de la saisie mais avec le formulaire contenant les informations déjà données au préalable. On peut aussi supprimer le constat. On a également la possibilité de consulter les informations précises d'un constat en cliquant sur le bouton « Informations du constat » à côté de chaque donnée.

Image 15 : Page de consultation d'un constat



Constats - Déclaratifs - Tableau de bord

Constat 2

Numéro : 2 Date de l'attaque : 2020-05-01 Date du constat : 2020-05-04 Nombre de victimes (morts) : 2 Nombre de victimes (blessés) : 6 Nom du 1er agent recenseur : agent 1 Nom du 2ème agent recenseur : agent 3 Propriétaire : Berger 2 Type d'animaux : 1 Secteur : Valbonnais Commune : Chantepérier Département : 38 Localisation : Dans le coeur Créateur du constat : test Administrateur Statut : 1 Nombre de jours agent : None



La carte est cette fois placée sur le lieu du constat à un gros niveau de zoom. On a aussi l'ensemble des informations du constat sur la gauche de la carte. On a aussi la possibilité de sauver sa carte au format image depuis cette page. C'est l'icône sous le contrôle du zoom qui permet l'export au format png.

Maintenant, il faut évoquer une fonctionnalité demandée par la commanditaire du projet et ses supérieurs qui est la réalisation d'un tableau de bord. Ce tableau doit donner rapidement des informations sur le nombre de constats et le nombre de victimes dans chaque secteur ou dans le cœur du PNE. L'image ci-dessous présente le tableau de bord de l'application GeoConstats.

Image 16 : Page de tableau de bord

Constats	 Déclaratif 	s 🔻 Table	au de bord							test Administra
Afficher les filtres Nombre de c département	onstat	s par								
Nombre de constats	38	05	Total							
Constats	5	1	6							
Déclaratifs	0	3	3							
Total	5	4	9							
Exporter le tableau Nombre de c	onstat	s par	secter	ur						
Exporter le tableau Nombre de c Nombre de constats	onstat _{Vallouise}	s par	secter	u r Igaudemar	Valbonnais	Oisans	Briançonnais	Embrunais	Total	
Exporter le tableau Nombre de c Nombre de constats Constats	Onstat Vallouise	S par Champs	secter saur Va 1	U l' Igaudemar	Valbonnais 2	Oisans 2	Briançonnais 0	Embrunais 0	Total 5	
Exporter le tableau Nombre de c Nombre de constats Constats Déclaratifs	Onstat Vallouise	S par Champs 0 0	secter saur Va 1 0	U l' Igaudemar	Valbonnais 2 0	Oisans 2 0	Briançonnais O O	Embrunais 0 0	Total 5 3	

Pour cela, je reprends le formulaire de filtrage de la carte auquel je joins des tableaux. En plus de cela, tous les tableaux sont exportables au format csv. Les données exportables ici sont donc des tris à plat là où celles sur l'interface de visualisation sont des données individuelles.

Avec le développement de ces fonctionnalités, nous n'avons pas encore parlé de la manière dont je vais appliquer mon diagramme de cas d'utilisations à l'application. Et c'est à cet endroit qu'intervient UsersHub. Je commence par charger un sous-module d'authentification de UsersHub (https://github.com/PnX-SI/UsersHub-authentification-module) disponible sur un dépôt Github. Ce module me permet de charger des routes Python semblables à celles que j'ai développé mais qui servent à se connecter à une application et vérifier les droits des utilisateurs pour la réalisation de certaines actions.

Afin de connecter UsersHub et GeoConstats, je me suis basé sur plusieurs applications utilisant UsersHub comme GeoPaysage (https://github.com/PnX-SI/GeoPaysages), TaxHub (https://github.com/PnX-SI/TaxHub) et UsersHub (https://github.com/PnX-SI/UsersHub) luimême. La majorité de ces projets utilisent l'objet Blueprint. Cet objet Blueprint représente une extension d'une application. On construit des routes de la même manière sauf que, au chargement de l'application, on peut appliquer un préfixe commun à toutes les routes d'un même fichier dans l'url. Le module d'authentification de UsersHub fonctionne aussi de cette manière (son préfixe d'url étant /api/auth). L'idée ici est de créer un objet Blueprint par fichier de routes puis d'appeler ces objets dans le fichier de lancement de l'application et éventuellement leur donner un préfixe pour différencier les multiples routes.

Pour vérifier si un utilisateur a bien le droit d'accéder à une page, on va comparer son niveau de droits avec le niveau requis juste avant l'exécution de la route se cachant derrière la page demandée. Si le niveau n'est pas suffisant, la route demandée ne s'exécute pas et un message d'erreur indiquant que les droits sont insuffisants apparait. En plus de cela, la page de visualisation ne va afficher les boutons de modification et de suppression d'un constat uniquement si les droits de l'utilisateur connecté sont suffisants.

Pour embellir l'application, j'ai également utilisé la librairie Bootstrap qui me permet d'appliquer du style sur les éléments de mes pages web juste en donnant une classe à un élément. L'image cidessous présente un exemple simple de ce que Bootstrap a apporté à GeoConstats.

Image 17 : Exemple d'utilisation de Bootstrap

Filtrer les données

Télécharger les données

Ce résultat pour des boutons est facilement obtenable et intègre parfaitement mon application. Maintenant que les fonctionnalités sont toutes expliquées, il est temps de parler du déploiement de GeoConstats.

2.2.3. Déploiement sur un serveur

Avant de parler du déploiement, il faut parler des deux stades des lesquels une application peut se trouver. Elle peut être soit en développement soit en production. Une application en développement peut se lancer très facilement et ne tourne pas en continu pour continuer le développement de celle-ci. Au contraire, une application en production est une application terminée qui va tourner de manière continue sur un serveur.

Pour développer GeoConstats, je ne me suis pas occupé de la version de production avant d'avoir fini complètement l'application. Une fois le développement terminé, mon maitre de stage m'a donné une place sur un serveur pour déposer mon application et ainsi pouvoir transformer mon application en développement en une application en production. Pour cela, j'ai dû ajouter un module complémentaire à Flask qui s'appelle Gunicorn. Gunicorn est un serveur WSGI qui vient donc assurer la communication entre le serveur web (Flask) et mon application web. Pour faire tourner toute cette architecture en arrière-plan sur le serveur, je dois également utiliser supervisorctl. Dans le cas de l'application, supervisorctl permet de relancer l'application dès que le serveur doit redémarrer pour une quelconque raison. Il me faut également un serveur web et je le dirige vers Apache. C'est Apache qui va gérer toutes les requêtes envoyées par l'utilisateur.

Après avoir installé mon application sur un serveur avec les outils évoqués dans cette partie et avoir exécuté le fichier install_db.sh évoqué en partie 2.2.1, je peux maintenant aller sur mon application depuis n'importe quel ordinateur disposant d'une connexion Internet grâce à l'URL suivante : http://178.32.193.149/geoconstats .

En plus de ce déploiement, il me faut documenter l'application pour que ses futurs administrateurs et utilisateurs puissent s'en servir. Pour la documentation d'administrateur, Github propose un fichier readme.md (https://github.com/PnEcrins/GeoConstats/blob/master/README.md) accessible sur la page d'accueil du dépôt de GeoConstats. Ce fichier servira de manuel administrateur pour l'installation de l'application. Pour le manuel utilisateur, j'ai rédigé un guide complet disponible dans les annexes de ce document.

3. Discussion des résultats

Le projet GeoConstats est complètement réalisé. Il permettra aux agents du Parc d'avoir une information claire sur la saisie des constats. Le stage s'est terminé par une présentation de l'application au personnel du PNE présent ce jour-là incluant le directeur. De plus, la mise à disposition de l'interface de visualisation au grand public serait un plus pour le Parc national des Ecrins dans un contexte très tendu entre les bergers et les protecteurs des loups.

Le loup est réapparu officiellement dans les Alpes françaises en 1992 dans le Parc national du Mercantour (certaines traces historiques laissent penser à un retour à la fin des années 80). Il a ensuite remonté les Alpes vers le Nord et est aussi parti à l'Ouest dans le Massif Central et les Pyrénées. Le loup est une espèce protégée depuis 1993. Concernant le loup dans le Parc national des Ecrins, c'est un phénomène très récent car il n'est revenu dans le Parc qu'il y a dix ans environ et sa présence est devenue régulière dans beaucoup de vallées depuis cinq ans. Aujourd'hui, toutes les vallées du PNE sont affectées par les attaques de loups, la dernière étant la vallée du Vénéon dans l'Oisans (2018).

L'administration française, de son côté, favorise la réintroduction du loup dans son milieu naturel. De plus, à l'intérieur du cœur d'un parc national, il est interdit de tirer sur un loup. Cependant, dans le cadre de la défense d'un troupeau d'animaux menacés par une attaque, les bergers hors du cœur du Parc peuvent tirer sur une bête mais ils doivent respecter un quota de 100 animaux tués par an. A titre d'exemple dans les Hautes-Alpes en 2017, on compte 1732 chasseurs habilités à tuer le loup.

Dans l'actualité récente, nous pouvons parler du film « Marche avec les Loups » de Jean Michel Bertrand qui a suivi plusieurs meutes de loups afin de voir comment un louveteau est éduqué à la vie dans sa meute et étudier ses déplacements. Son film a amené beaucoup de réactions chez les bergers qui ont l'impression que l'on favorise le loup à leurs activités. Entre autres, une manifestation était prévue à Gap le jour d'une avant-première du film avec l'idée de distribuer des tracts présentant le quotidien des bergers à cause des loups.

Avec les tensions présentées ici, le Parc national des Ecrins souhaite que les données liées aux loups soient présentées au grand public en consultation. Le projet GeoConstats présente donc un double atout car il présente les données des constats liés aux attaques de loups et simplifie la saisie de ces données.

Pour ce qui est de la demande, le contexte de télétravail a ralenti l'activité de mon stage, ce qui a amené mon maître de stage et moi-même à revoir nos objectifs en fonction de ce qui était réalisable. La solution la plus pratique a été choisie avec un projet adaptable à un environnement de télétravail et une première base apportée juste avant le confinement. Ensuite, pour l'application, le schéma suivi est classique avec d'abord une analyse des besoins et de l'existant avant de schématiser l'application. La suite est un développement étape par étape pour petit à petit ajouter des fonctionnalités à l'application. Le seul point ayant été un problème concerne la base de données qui a évolué au fur et à mesure du projet pour ajouter des champs aux constats ou pour ajouter de nouvelles tables permettant de réaliser de nouvelles fonctionnalités. Cela m'a amené de manière fréquente à reprendre beaucoup de fichiers pour ajouter les champs supplémentaires. Le projet de restructuration du SIG a été laissé en cours à la fin de sa première phase mais la seconde phase aurait été perturbée par le retard pris dans la refonte du site marchand de l'IGN. Ce site devait nous fournir des fonds de carte ainsi que de nouvelles bases de données. Pour le projet sur les données de climatologie, il a été laissé de côté mais le code de l'application GeoConstats est réutilisable pour cette application aux besoins similaires.

4. Conclusion

Le projet initial du stage a été grandement modifié avec l'évolution des besoins et de mon contexte de travail mais le projet GeoConstats a vu le jour. Cette application sera utilisée à partir de 2021 pour ne pas perturber les agents en pleine saison de saisie de constats. Seuls quelques agents volontaires basculeront sur GeoConstats cette année afin de servir de test pour mon application. Malheureusement, je ne pourrai pas m'occuper des retours des tests car ils commenceront après mon départ. Cependant, la documentation réalisée pour l'application devrait permettre à mon maître de stage de très vite maitriser complètement l'application. De plus, une idée a été évoquée pour une future évolution de l'application qui s'occuperait des attaques visant les chiens protégeant les troupeaux. Le projet GeoConstats servira aussi de base pour la réalisation du projet de mise en valeur des données météorologiques (GeoClimat).

Ce stage m'aura fait utiliser l'ensemble de mes compétences apprises au sein du Master Géomatique avec beaucoup de programmation. Sur cet aspect, le stage a approfondi des notions vues en cours et abordé certains points plus avancés mais très utiles. Pour la partie SIG (détaillée en annexe), elle a grandement été réduite à cause du télétravail mais j'ai tout de même pu mettre en valeur ces compétences pendant un mois. De plus, j'ai pu manipuler QGIS dans un environnement Linux-Ubuntu qui est complètement différent d'un environnement Windows et je n'avais aucune connaissance de ces changements en fonction du système d'exploitation. Mais ce stage a avant tout continué le développement de mes compétences en programmation avec un projet utilisant des technologies abordées durant le Master Géomatique (Flask, Leaflet, PostgreSQL). D'un point de vue méthodologique, j'ai pu me rendre compte que le système de point quotidien entre tous les membres du pole SI m'a vraiment permis de trouver des solutions à mes problèmes de manière efficiente. Principalement pendant le télétravail, ces points quotidiens en visio-conférence m'ont permis de ne pas rester bloqué sur un objectif très longtemps.

Cette expérience a été très enrichissante sur un territoire que je ne connaissais que partiellement (2 vallées sur 7). J'ai pu également obtenir beaucoup d'informations sur les projets menés par le Parc national des Ecrins et de manière générale sur l'ensemble des parcs nationaux.

5. Bibliographie

1) Site regroupant des données géologiques du PNE : <u>https://geologie.ecrins-parcnational.fr/</u>

2) Dépôt Github du PNE : <u>https://github.com/PnEcrins/</u>

3) Dépôt Github des pôles SI des parcs nationaux de France : <u>https://github.com/PnX-SI/</u>

4) Maploup : <u>https://maploup.fr/</u>

5) Marche avec les loups en avant-premières, Parc national des Ecrins, 12 septembre 2019 http://www.ecrins-parcnational.fr/actualite/marche-loups-premieres

6) Le retour du loup désespère les bergers, Ouest France, <u>https://www.ouest-</u> france.fr/economie/agriculture/le-retour-du-loup-desespere-les-bergers-6914650

7) La sortie sous haute tension du film "Marche avec les loups", Anne-Sophie Tassart, 10 janvier 2020 <u>https://www.sciencesetavenir.fr/animaux/biodiversite/avant-la-sortie-du-film-marche-avec-les-loups-le-monde-pastoral-serre-les-machoires_140377</u>

8) Le retour du loup, Parc national des Ecrins, <u>https://www.paysdesecrins.com/experiences/news-</u> <u>du-parc-national-des-ecrins/le-retour-du-loup</u>

6. Glossaire

DDT : Direction Départementale du Territoire

IGN : Institut Géographique National

OFB : Office française de la Biodiversité

ORM : Object Relational Mapper

PNE : Parc national des Ecrins

SIG : Système d'informations géographiques

WSGI: Web Server Gateway Interface

7. Annexes

7.1. Projet de modernisation du SIG

Le projet de modernisation du SIG consiste à investiguer sur le format des fichiers contenant l'information géographique du Parc national des Ecrins. A mon arrivée, les fichiers contenant des données vectorielles sont au format shapefile. Cependant, mon maître de stage aimerait s'en débarrasser pour un format plus simple d'utilisation. C'est dans cette optique que j'ai démarré une analyse comparative entre trois formats de données potentiels : le shapefile, le Géopackage et le GeoJSON. L'analyse comporte des tests sur l'ouverture et l'écriture d'un fichier dans QGIS. On ouvre le fichier « en local » et sur un serveur. On étudie aussi le comportement sur différents systèmes d'exploitation (Windows, Ubuntu). Après avoir réalisé cette étude, j'ai créé un (https://github.com/PnX-SI/Ressourcesprogramme en Python techniques/tree/master/Python/Shp2GPKG) pour transformer chaque fichier shapefile d'un dossier et de ses sous-dossiers au format GeoPackage. La réalisation de ce script m'a fait apprendre à utiliser un nouveau module de Python qui est Fiona. Pour finir, j'ai rédigé un article en HTML mis à la disposition des pôles SI des autres parcs nationaux français en plus du script Python. Le script Python a déjà été utilisé par le Parc national de la Vanoise. Pour détailler les étapes de ce projet, la suite de cette partie est l'article que j'ai rédigé à la destination des autres parcs nationaux. La suite de cette partie est cet article (https://si.ecrins-parcnational.com/blog/2020-02-geojsonshapefile-geopackage.html).

GeoJSON VS Shapefile VS GeoPackage

Raphaël Bres - Parc national des Ecrins / Février 2020

1. Introduction

Le Parc national des Écrins est organisé avec un siège à Gap et 7 antennes sur le territoire, localisées dans des maisons de parc dans les vallées. Plus de 40 agents manipulent des données géographiques avec le logiciel libre QGIS. Le SIG est organisé avec des couches de référence partagées en lecture et des données thématiques créées par les agents.

Le Parc national des Écrins souhaite trouver un format de données pour remplacer l'ensemble des couches vectorielles au format shapefile. La solution idéale de se baser sur une base de données PostGIS a été écartée à cause de l'éclatement des 8 sites de travail et de leur débit internet faible et irrégulier.

Nous avons donc étudié deux alternatives que sont le GeoJSON et le GeoPackage. Commençons par définir ces trois types de données avant de présenter les résultats de tests de performance et de conclure sur le format retenu.

2. Les différents formats de données

Le format shapefile est le format le plus démocratisé de nos jours dans l'utilisation des SIG. Il a pendant très longtemps été la recommandation principale de l'OGC. Cependant, ce format dispose de plusieurs inconvénients comme la création de plusieurs fichiers pour une même couche (SHP, DBF, SHX, PROJ). De plus, ce format est réputé très lourd et est limité à 2 Go de stockage par fichier shapefile.

Le format GeoJSON est un format non préconisé par l'OGC pour traiter l'information géographique mais est largement utilisé pour du développement web à travers des



librairies comme Leaflet, OpenLayers ou MapboxGL. Il est dit comme très performant concernant de petits jeux de données mais a de grosses limites lorsque le nombre de données augmente.

Le format GeoPackage est le format le plus récent recommandé par l'OGC pour les traitements sur les SIG. Il est par exemple le format de données par défaut de QGIS3. Ce format est basé sur une base de données SQLite qui est censée accélérer les requêtes spatiales et le temps d'affichage de la couche, principalement quand on a de gros volumes de données. Sa principale contrainte vient du fait que sa structure n'est pas adaptée à une mise à disposition sur un serveur.

Voir http://switchfromshapefile.org pour plus de détails sur ces 3 formats.

3. Test des formats

Maintenant que nous avons présenté les données, nous allons tester l'ensemble de ces formats pour savoir lequel est le plus performant. Pour cela, nous allons prendre un fichier « léger » avec environ 250 données et un fichier « lourd » avec plus de 600 000 données. Nous allons d'abord comparer la taille de ses fichiers avant de charger dans QGIS 10 fois chaque fichier afin d'avoir des données précises sur le temps de chargement de chaque format.

D'abord, évoquons la taille de chaque fichier selon son format. Ici, nous allons définir quel format de données prend le moins de place à stocker.

Tableau de comparaison des tailles de chaque fichier

	GeoJSON	SHP	GPKG
Fichier léger	60Ko	50Ko	135Ko
Fichier lourd	400Mo	2Go	200Mo

Dans ce cas, nous voyons que le format Shapefile est beaucoup plus volumineux que les autres lorsque l'on utilise un fichier lourd. La taille du fichier lourd est divisée par 5 pour le même fichier en GeoJSON et est divisée par 10 pour le format GeoPackage. Par contre, pour les fichiers légers, nous voyons que le GeoPackage est 2 fois plus lourd que les autres du à sa structure complexe. Dans ce cas, le shapefile est le fichier le plus léger. Pour conclure sur le volet de la taille, vu que le PNE dispose de gros fichiers de données, sur la faune et la flore par exemple, nous privilégions le GeoPackage du point de vue de la taille des fichiers.

Maintenant, nous passons au test dans QGIS. Il est important de noter que les temps présentés ici peuvent varier d'une machine à une autre. Le test compare le temps d'ouverture (en secondes) d'un fichier léger et d'un fichier lourd, stocké en local sur le PC et sur le serveur de fichiers. On présente ici le ratio entre deux temps de chargement.

Taille d	u fichier	Fichier léger									
Local/	Serveur		Local		Serveur						
Exte	nsion	GeoJSON vs SHP	GeoJSON vs GPKG	SHP vs GPKG	GeoJSON vs SHP	GeoJSON vs GPKG	SHP vs GPKG				
Ubuntu	Ouverture	1,150	0,744	1,545	0,813	0,234	3,477				
Windows	Ouverture	0,706	1,050	0,672	0,905	0,927	0,976				
	Ecriture			Insta	ntané						

Résultats de 10 itérations pour différents formats

Interprétation : Sur un système Ubuntu, pour ouvrir un fichier, le format SHP est 1,15 fois plus rapide que le format GeoJSON.

Taille d	u fichier	Fichier lourd									
Local/	Serveur		Local		Serveur						
Extension		GeoJSON vs SHP	GeoJSON vs GPKG	SHP vs GPKG	GeoJSON vs SHP	GeoJSON vs GPKG	SHP vs GPKG				
Ubuntu	Ouverture			0,908			2,353				
Windows	Ouverture	7,450	31,561	4,236	3,568	4,267	1,196				
	Ecriture	37,046	58,950	1,591	0,190	2,120	11,129				

Nous constatons que le GeoPackage présente des avantages indéniables quand on utilise un très grand nombre de données. Principalement en écriture de données, il est 11 fois plus rapide que le format shapefile alors que le format GeoJSON est lent à l'ouverture. Le problème majeur du format GeoJSON reste son inefficacité sur un poste avec un système d'exploitation Linux où l'ouverture d'un gros fichier de données entraîne le plantage de QGIS.

4. Conclusion

Pour conclure cette analyse, le format Shapefile peut être laissé de côté par rapport au format GeoPackage qui dispose de meilleures performances et prend moins de place lorsque l'on souhaite stocker de gros fichiers.

Pour automatiser la conversion de tous nos fichiers SHP en GeoPackages, j'ai développé un <u>script Python dédié</u> (<u>https://github.com/PnX-SI/Ressources-techniques/tree/mas-ter/Python/Shp2GPKG</u>).

7.2. Compte rendu de la réunion de lancement du projet sur l'agriculture et le pastoralisme

Base de données agriculture : Définition des besoins

11 mars 2020 Participants : Muriel Della-Vedova, Camille Monchicourt, Raphaël Bres

Rédacteur : Raphaël Bres

Deux besoins sont clairement exprimés pour les données d'agriculture et de pastoralisme :

• Un besoin général pour mettre à disposition des données utiles aux agents et que ceux-ci pourraient modifier en direct

• Un besoin pour la gestion des constats et des attaques de loups. L'idée simple serait qu'un agent dispose d'une interface pour saisir un point et rentrer deux ou trois informations supplémentaires sur le constat.

Actuellement, les données sur les attaques de loups sont gérées par la DDT mais elles sont mal structurées et ne contiennent pas de localisation précise. Il existe également une plateforme nommée GeoLoups qui répertorie les attaques mais il manque environ un tiers des données donc on ne peut pas se baser sur cette application aux données incomplètes. Ce projet d'inventaire des attaques de loups serait uniquement en interne pour la saisie des constats par les agents. Cela facilitera la saisie des données et apporterait une plus grande précision dans la localisation de l'attaque car actuellement, la DDT ne donne qu'un lieu-dit pour localiser le constat et cela ne s'applique que dans le meilleur des cas. Pour la saisie, elle s'effectuerait en même temps que celle pour la DDT avec une éventuelle validation à posteriori si il s'agit bien d'une attaque de loups. Il faut également tenir compte des constats que peuvent créer les bergers en mode déclaratif si le constat recense moins de cinq victimes. Avant de penser à un tel développement, il faudra avoir l'aval des chefs de secteurs pour avoir un avis des futurs utilisateurs de cette application. Il faudra aussi localiser les constats déclaratifs à partir des unités pastorales. Il est aussi important de noter que la demande de ce projet est plus centrée sur le cœur du parc national que sur l'ensemble de l'aire d'adhésion. Pour le projet de publication de données liées au pastoralisme, celui-ci doit d'abord être précédé d'une mise à jour des données pour faire le tri entre ce qui est nécessaire de publier, ce qui est nécessaire uniquement en données SIG, ce qui peut être archivé et ce qui peut être supprimé. Une fois cette étape réalisée, on pourra construire un modèle de projet QGIS afin de publier ces données via un projet LizMap accessible sur internet à tous les agents du PNE. Ce projet ressemble beaucoup à celui réalisé par le Parc national du Mercantour (pour lequel nous avons eu accès à leur cartothèque LizMap sur un serveur de développement) cependant, nous n'allons pas publier les mêmes données car il y a certaines différences géographiques entre ces deux parcs. Pour aborder les données, nous avons d'abord les unités pastorales et les zones pastorales qui servent de délimitations dans ces données.les unités sont utilisées l'été et le reste du temps on utilise les zones pastorales. On obtient cette donnée depuis l'enquête pastorale réalisée par l'INRAe en 2014. Les thèmes abordés grâce à cette application seront : les unités pastorales, les zones pastorales, les cabanes pastorales et héliportables, les prairies de fauche, les lignes de lecture, les pédiluves et les limites administratives du parc national. La suite de ce projet sera de la schématisation de base de données pour y voir beaucoup plus clair au sein de toutes ces données.

7.3. Manuel utilisateur

GeoConstats : manuel utilisateur

L'application GeoConstats permet le suivi, l'ajout, la mise à jour et la suppression des constats d'attaques de loups. Dans ce manuel, nous allons voir comment utiliser cette application du point de vue d'un agent recenseur de constats.

Lorsque l'on arrive sur l'application, on se retrouve sur la page de connexion. Les identifiants de connexion sont ceux pour accéder à vos applications traditionnelles comme la boite mail par exemple. L'image ci-dessous montre la page de connexion à GeoConstats.

Page de connexion à GeoConstats

GeoConstate	6
agent	
•••••	
Connexion	

Après s'être connecté, on se retrouve sur la page d'accueil qui est la page de visualisation des constats. Cette page est présentée dans la capture d'écran ci-dessous.

Page de visualisation des constats



On retrouve une barre de navigation en haut de la page pour accéder aux pages sur les constats, sur les déclaratifs ainsi que le tableau de bord, le site du par cet le bouton de déconnexion. Ensuite, on a la carte qui occupe la majeure partie de l'écran. On a toutes les données de constats dessus. Sur la gauche de la carte, on a un formulaire pour filtrer les données ainsi qu'une liste des constats présents sur la carte (la liste dépend du formulaire, pas de la carte). Nous allons maintenant détailler les actions possibles avec la carte. On commence par zoomer (soit avec la molette de la souris, soit avec le bouton + en haut à gauche de la carte) sur la zone concernée. Pour l'exemple, je vais zoomer sur Bourg d'Oisans.

Page de visualisation des constats, zoom sur le nord de Bourg d'Oisans



En cliquant sur une punaise, on accède aux informations du constat localisé en ce lieu. Le rond bleu avec un 2 à l'intérieur signifie ici que deux constats sont placés au même endroit. On peut accéder aux données de chaque point en cliquant sur ce rond. Mais on peut aussi accéder aux données en trouvant le constat dans la liste à gauche de la carte. La date marquée est celle de l'attaque. L'image ci-dessous montre les données affichées pour un constat.

Visualisation des données dans la liste



On retrouve ici toutes les informations du constat. On peut également voir trois boutons sous les informations du constat. On peut modifier, supprimer et visualiser les informations du constat. Les boutons de modification et de suppression ne sont visibles que sur les constats que l'utilisateur a lui-même créé (pour les administrateurs et pour les référents, ils peuvent faire toutes les actions sur tous les constats. On clique sur le bouton « Informations ». On obtient la page présentée ci-dessous.

Page d'informations d'un constat

On retrouve les informations du constat sur la gauche et la carte avec seulement le constat dont

Numéro : 72 Date de l'attaque : 2020-07-24 Date du constat : 2020-07-24 Nombre de victimes (morts) : 2 Nombre de victimes (blessés) : 0 Nom du 1er agent recenseur : Prénom Nom Nom du 2ème agent recenseur : Prénom Nom Propriétaire : Prénom Nom Type d'animaux : 1 Secteur : Valbonnais Commune : Entraigues Département : 38 Localisation : Dans l'aire d'adhésion Créateur du constat : test Agent Statut : 1 Nombre de jours agent : 1.0

Or terins



on veut les informations. Sous le bouton de zoom sur la carte (en haut à gauche), on a un bouton permettant de télécharger la carte au format d'image.

Ensuite, on revient sur la page de visualisation des constats et on clique sur le bouton « Modifier ». Cela nous amène sur la page présentée ci-dessous.

Page de modification d'un constat



On dispose sur cette page d'une carte avec le constat positionné et sur la gauche, le formulaire avec les informations actuelles du constat déjà remplies. Une fois que l'on a fait les modifications que l'on souhaite, on clique sur le bouton tout en bas du formulaire pour valider les changements et on revient sur la carte de visualisation des constats. Sue cette page de visualisation des constats on peut aussi supprimer un constat (uniquement les constats que l'utilisateur a lui-même créé). Attention, juste un clic amène la suppression définitive du constat.

On peut aussi trouver un constat précis en utilisant le formulaire de filtrage des données. On peut filtrer les données par année de l'attaque, type d'animaux victimes de l'attaque, statut du constat, localisation (cœur du PNE, aire d'adhésion), commune et secteur. L'image ci-dessous présente le formulaire rempli juste avant de le valider.

Page de visualisation des constats, formulaire de filtrage



On clique sur le bouton « Filtrer les données ». La page de visualisation se recharge mais seulement avec les données correspondant à l'ensemble des filtres demandés. La liste des constats n'affiche que ces données filtrées.

Page de visualisation des constats après filtrage



Le bouton « Télécharger les données » permet d'exporter les données de constats au format csv (ouvrable avec un tableur). Les données téléchargées correspondent aux filtres présents dans le formulaire au moment du clic sur le bouton de téléchargement.

Ensuite, on peut également ajouter des constats en se servant de la barre de navigation. En allant comme montré dans l'image ci-dessous, dans « Constats » puis « Ajouter un constat » on peut ajouter un nouveau constat sur l'application.

Localisation du lien pour accéder à la page d'ajout d'un constat



Une fois avoir cliqué sur ce lien, on arrive sur la page permettant d'ajouter un constat.

Page de saisie d'un nouveau constat



La saisie se passe en deux temps. D'abord, on commence par placer le constat sur la carte. On peut se déplacer et zoomer sur la carte comme sur la page de visualisation et, lorsque l'on clique sur la carte, une punaise apparait indiquant que le point est bien pris en compte. On peut maintenant passer à la deuxième partie qui concerne le formulaire. On rentre les informations comme indiqué demandé et on valide avec le bouton en bas du formulaire. Il est important de noter que tant que le constat n'est pas localisé, le bouton de validation d'un constat n'est pas utilisable. L'image ci-dessous montre un exemple d'un ajout de données juste avant de valider le constat.

Page d'ajout d'un constat juste avant la validation



Une fois le constat validé, on retourne sur la page de visualisation des constats. On retrouve le constat ajouté sur la carte et dans la liste.

Pour les constats déclaratifs, le fonctionnement est le même que pour les constats. Si vous avez un problème en manipulant les constats déclaratifs, cherchez comment réaliser cette action pour un constat.

Pour finir, parlons du lien « Tableau de bord » dans la barre de navigation. En cliquant dessus, on accède au même formulaire de filtrage que sur la carte avec des tableaux donnant le nombre de constats par département, le nombre de constats par secteur et le nombre d'animaux tués par secteur. On peut aussi télécharger les tableaux séparément avec les filtres appliqués. L'image ci-dessous montre le tableau de données en lui ayant appliqué des filtres.

Page de tableau de bord

Constats - Décla	ratifs 👻 Table	au de bord Sit	e du Parc	C						test Agent 👻
Filtres sur les constat	ts									
Année du constat 2020 v Type d'animaux Ovins v Statut du constat Indemnisé v Fibrer les données	ats par (départe	ment	t						
Nombre de constats	05	38	Total							
Constats	8	0	8							
Déclaratifs	6	0	6							
Total	14	0	14							
Exporter le tableau Nombre de const	ats par :	secteur								
Nombre de constats	Vallouise	Champsau	ır	Valgaudemar	Valbonnais	Oisans	Briançonnais	Embrunais	Total	
Constats	0	2		2	0	0	0	4	8	
Déclaratifs	1	3		0	0	0	0	2	6	
Total	1	5		2	0	0	0	6	14	
Exporter le tableau Nombre d'animau	ıx morts	par seo	cteur							

Pour finir, vous pouvez vous déconnecter en cliquant sur votre nom d'utilisateur en haut à droite puis sur le bouton « Déconnexion ».

8. Résumé

8.1. En français

Les missions que je dois accomplir pour le Parc national des Ecrins sont la structuration, la diffusion et la mise en valeur de données géographiques. Cela consiste à moderniser leur système de stockage de données, mettre en ligne certaines données sur des plateformes d'Open Data et créer des applications web présentant des données sur une thématique précise. Ce travail est d'abord à la destination des utilisateurs de données géographiques. Pour réaliser ces missions, je vais principalement utiliser des logiciels et langages de programmation libres et open source comme QGIS, PostgreSQL ou encore Python. Afin de répondre à cette commande, je vais d'abord réaliser la partie de modernisation du SIG car elle me permet de mieux connaître les données du Parc national des Ecrins. Ensuite, je réaliserai la mise en valeur de certaines données qui est le projet le plus volumineux de ce stage.

8.2. En anglais

The missions I have to do for Parc national des Ecrins are geographical data structuring and diffusion. It is the modernization of their data storage system, publish some data on Open Data platform and create web apps presenting data on a precise theme. This task is to simplify the daily work of some employees by making some data more accessible. To do those missions, I will mainly use free and open source software or programming languages like QGIS, PostgreSQL, or Python. I will begin by the GIS modernization part because this part will make me know the data I will use. After this, I will make some web apps to highlight some data that were not treated a lot in the past.