

**Sujet : Orpellières 1**  
**Équipe 8**

## **PARTIE COLLECTIVE**

# **Introduction**

Pendant le hackathon, nous avons été commandités par la Communauté d'agglomération Béziers Méditerranée – Département de l'aménagement et de la transition écologique pour répondre à ces problématiques :

Comment améliorer

- la maîtrise de la fréquentation sur le site naturel protégé des Orpellières
- la visibilité numérique des Orpellières et de nos missions
- Suivre et évaluer l'évolution des habitats naturels (graus, lagunes, trait de côté et cordon lunaire)

Pour pouvoir y répondre, nous avons décidé comme approche de créer une base de données (la récolte des données se fait automatiquement) afin de la lier avec le reste du projet qui serait un site web.

## **Solution globale RT – Base de données**

Tout d'abord, comme nous n'avions pas à disposition de base de données complète fournie (seulement les analyses vélo/piéton) par Orpellières, nous avons décidé de nous appuyer sur des données en ligne sur d'autres sites, comme le site INPN Orpellières :  
<https://inpn.mnhn.fr/site/natura2000/FR9101434>

On a aussi constaté qu'il y avait des données qui changeaient souvent et d'autres non. Donc, on a décidé de transformer les données en deux bases de données : une partie administration et une partie biodiversité. Avec une troisième base de données pour les cyclistes/piétons.

Après, on a créé des scripts pour récupérer des images correspondant à chaque espèce pour permettre un meilleur affichage des données en utilisant soit l'API de Wikipédia, soit l'API de Unsplash.

Enfin, on a relié les bases de données au site web créé par les autres membres du groupe grâce à des scripts en JavaScript.

Pour finir, on a utilisé des technologies GitHub (GitHub Pages) pour mettre en service le serveur web ainsi que toutes les autres ressources comme les images (pour certaines vidéos, on a utilisé Youtube pour alléger le service).

# Rapport avec la formation

Ici, le travail est en lien avec l'UE programmée. En effet, on crée des bases de données à l'aide de scripts en python en les organisant automatiquement et en les réutilisant sur des pages web. De plus, on a utilisé GitHub et Visual Studio Code pour organiser notre espace de travail et se répartir les différentes tâches.

Pour résumer, on a utilisé plusieurs langages de programmation, créé de nombreux scripts se rattachant directement ou indirectement les uns aux autres pour récolter, analyser, etc., les données, fait de l'automatisation de tâches (récolte des données, recherche d'image). On a aussi utilisé GitHub, une application de travail de groupe, mis en lien nos bases de données avec le serveur web et mis en service ce serveur.

## Choix des services utilisés

Pour le travail de groupe, on a préféré utiliser GitHub et Visual Studio Code. Cela nous a permis de voir la progression de chacun tout en séparant les espaces de travail grâce aux branches sur GitHub.

GitHub nous a aussi permis de déployer les services nécessaires, comme le serveur web et les scripts. Et l'utilisation des workflows internes de GitHub nous a permis, à l'aide d'un fichier YAML exécutant des commandes bash, d'automatiser tous nos services. De la récolte de données jusqu'à la mise en place du serveur web.

Après, l'utilisation de Visual Studio Code (VS Code) nous a rendu la tâche très facile, car il fonctionne sur la majorité des OS, tout en permettant une compatibilité inter-OS. De plus, le fait que GitHub soit utilisable sur le terminal intégré de VS Code nous a grandement aidés à être plus efficaces dans le partage de nos travaux.

## Choix techniques

Pour les données que nous avons choisies, comme les données disponibles étaient très limitées, on a opté pour la récupération en ligne de données. Sur le site INPN Orpellières, on a pu retrouver beaucoup de données sous format XML qui est adapté à la récupération de données par scripts python.

En récupérant les données en ligne depuis le site INPN, nous avons dû trier les données pour ne garder que l'essentiel. Pour ce faire, on a dû créer deux scripts (partie admin, partie biodiversité).

On a aussi décidé de convertir sous forme de dictionnaire les données triées avant de les écrire dans un fichier (un pour chaque base de données) au format JSON, car il est très facile à comprendre et plus léger que XML. Par conséquent, plus rapide que XML.

Chaque fichier est écrit selon son mois et son année, car la récupération se ferait tous les mois (mise à jour du site INPN tous les 1-2 mois) et permettrait de conserver les anciennes données pour avoir l'évolution de celles-ci au fil du temps.

DEDIEU Tristan  
WILHEM Tom

## Solution Technique RT1 : Hackaton

Après la récupération des données, on a décidé de changer toutes les valeurs ambiguës en valeurs compréhensibles (G ==> bonne) car elles seraient celles qui seraient affichées pour l'utilisateur.

On a utilisé du JavaScript pour mettre en lien nos données sur le site web, car il était plus facile de le faire sur ce langage que sur d'autres (comme du python, etc.).

On a aussi créé des scripts Python permettant la récupération d'image en fonction de noms d'espèces pour simplifier l'affichage du site web de manière automatique (s'effectuera même si l'espèce n'était pas initialement dans l'Orpellières).

## PARTIE INDIVIDUELLE

# Compétences acquises

Pendant cette période de hackathon, j'ai pu utiliser la plupart de mes compétences de l'UE Programmer tout en apprenant de nouvelles compétences.

Pour commencer, j'ai eu l'occasion de travailler beaucoup avec Python, notamment l'écriture et l'analyse de fichiers XML/JSON.

En effet, j'ai l'occasion de créer des scripts Python permettant la récupération de données au format XML avant de les trier et de les écrire dans un autre fichier en JSON.

Ensuite, j'ai pu pour la première fois utiliser le service GitHub afin de travailler en groupe. Le tout, sur l'application Visual Studio Code.

J'ai par exemple pu utiliser des commandes comme :

- git add .
- git commit
- git push
- etc.

Afin de partager, récupérer les fichiers et répertoires présents dans les différentes branches.

J'ai aussi eu l'occasion de tester des librairies JavaScript et, par conséquent, de programmer dans ce langage.

Parmi ces librairies, j'ai pu utiliser Leaflet afin de voir comment créer une carte interactive.

# Difficultés rencontrées

Il y a pas mal de choses que je n'ai pas pu faire, soit car il nous manquait du temps, soit car je n'avais (ou n'ai) pas les compétences requises.

En effet, je n'ai utilisé que très peu GitHub avant ce projet, donc pour l'hébergement du site, j'ai dû me reposer sur Tom pour qu'il m'explique ce qu'il faisait et pourquoi.

DEDIEU Tristan  
WILHEM Tom

## Solution Technique RT1 : Hackaton

De plus, je n'avais pas programmé en JavaScript auparavant, donc j'ai eu énormément de mal à faire ce qu'il fallait. J'ai donc dû me limiter à beaucoup de tests sans forcément avoir un rendu concret.