|  |
| --- |
| RAPPORT PROJET TUTORE : LES LIAISONS MARITIMES DANS LE PACIFIQUE SUD |
|  |

|  |
| --- |
| DINAN Olivier – TRAM Olivier  Tuteur : SALMON Loïc |

# Partie 1 : Introduction

## Le Pacifique Sud

Le Pacifique Sud, s'étendant sur une vaste étendue de l'océan Pacifique. Celui-ci représente une région stratégique tant sur le plan économique que géopolitique. Cette zone comprend de nombreux pays insulaires, tel que la Nouvelle-Calédonie, la Polynésie française et les îles Salomon, chacun ayant des liaisons maritimes essentielles pour le commerce, le tourisme, et les échanges culturels. Les liaisons maritimes dans cette région sont cruciales pour la connectivité entre ces îles et les pays voisins, comme l'Australie et la Nouvelle-Zélande. Ces pays jouent un rôle vital dans le développement économique des nations insulaires, qui dépendent souvent de ces voies maritimes pour l'importation de biens, l'exportation de ressources naturelles et le tourisme. En outre, la région est confrontée à des défis tels que l'isolement géographique, la variabilité des conditions climatiques et les enjeux environnementaux, ce qui rend d'autant plus important le suivi et l'optimisation des routes maritimes.

## Problématique

Face à l'importance croissante des échanges maritimes dans le Pacifique Sud, une question cruciale émerge : comment analyser et visualiser efficacement les trajets des navires dans cette région dynamique ? Le manque de données centralisées et l'hétérogénéité des systèmes de suivi posent des défis significatifs pour les chercheurs, les gestionnaires de ports et les autorités maritimes. Une solution innovante est nécessaire pour répondre à cette problématique et améliorer la compréhension des flux maritimes

## Objectif

L'objectif principal de ce rapport est de développer un outil de suivi des trajets maritimes, basé sur des données récupérées via des techniques de scrapping et visualisées sur une carte interactive. Cet outil vise à offrir une plateforme accessible pour analyser les mouvements des navires dans le Pacifique Sud, facilitant ainsi la prise de décision pour les acteurs concernés.

## Méthodologie

Pour atteindre cet objectif, nous avons adopté une méthodologie combinant plusieurs technologies modernes. Nous avons utilisé le langage Python pour le développement de scripts de scrapping afin de récupérer des données sur les trajets maritimes à partir de sources en ligne et d’api. Les données extraites sont ensuite stockées dans une base de données SQL pour une gestion efficace. Enfin, nous avons employé Folium qui est une bibliothèque Python, pour créer une carte interactive qui permet de visualiser les trajets des navires en temps réel. Cette section du rapport décrira en détail les différentes étapes de notre démarche et les choix technologiques effectués.

# Partie 2 : Préparation au projet

## L’Automatic Identification System

L'Automatic Identification System (AIS) est une technologie de suivi maritime qui permet de suivre les navires en temps réel. Développé à l'origine pour améliorer la sécurité maritime, l'AIS utilise des signaux radio VHF pour transmettre des informations cruciales sur la position, la vitesse, et la direction des navires. Chaque navire équipé d'un transpondeur AIS envoie régulièrement des messages contenant ces données, qui peuvent être reçus par d'autres navires et par des stations à terre.

L'AIS joue un rôle essentiel dans la surveillance des trajets maritimes, notamment dans les eaux congestionnées et les ports. Il contribue également à la prévention des collisions et facilite la gestion du trafic maritime. Cependant, bien que l'AIS soit une source précieuse d'informations, il présente des limites. Par exemple, les navires de petite taille ne sont pas toujours équipés de transpondeurs AIS, ce qui peut créer des lacunes dans les données disponibles. De plus, la portée des signaux VHF est limitée, ce qui peut affecter la couverture dans certaines zones éloignées du Pacifique Sud.

## Obtention de données

Avant de poursuivre, il est important de définir certains termes afin de bien comprendre les éléments essentiels de notre projet :

* MMSI : Identifiant unique attribué à chaque bateau.
* Latitude : Coordonnée géographique représentant la position verticale.
* Longitude : Coordonnée géographique représentant la position horizontale.
* Speed : Vitesse du bateau en kilomètres par heure.
* Received\_at : Date et heure de réception des données.
* Shipname : Nom du bateau.
* Shiptype : Type de bateau (ex. cargo, bateau de pêche, etc.).

Les termes ont été défini par notre professeur, qui nous a fourni des données dans le format csv pour pouvoir commencer.

# Partie 3 : Technologie Utilisé.

## Langages de programmation

### **Python**

Python est un langage de programmation polyvalent et facile à apprendre, largement utilisé dans le développement web, l'analyse de données, le machine learning et le scrapping. Pour notre projet, nous avons utilisé plusieurs bibliothèques Python spécifiques :

* **CSV** : Cette bibliothèque intégrée permet de lire et d'écrire des fichiers au format CSV (Comma-Separated Values). Elle est particulièrement utile pour manipuler des données tabulaires, facilitant l'importation et l'exportation de données entre différents formats.
* **Psycopg2** : Il s'agit d'un adaptateur PostgreSQL pour Python. Il permet de se connecter à des bases de données PostgreSQL, d'exécuter des requêtes SQL, et de gérer les transactions. Cette bibliothèque est essentielle pour stocker et manipuler les données récupérées.
* **AsyncHTMLSession** : Faisant partie de la bibliothèque requests-html, AsyncHTMLSession permet d'effectuer des requêtes HTTP de manière asynchrone, facilitant le scrapping de plusieurs pages web simultanément. Cela améliore considérablement la vitesse de collecte des données.
* **BeautifulSoup** : C'est une bibliothèque de scrapping HTML qui permet d'extraire des données de documents HTML et XML. Elle facilite l'analyse de la structure d'une page web et l'extraction d'informations spécifiques, rendant le processus de scrapping plus efficace.
* **Asyncio** : Cette bibliothèque est utilisée pour écrire du code concurrent en Python. Elle permet de gérer des opérations asynchrones, telles que des requêtes HTTP, ce qui est particulièrement utile lors de la collecte de données sur plusieurs sources simultanément.
* **Folium** : Folium est une bibliothèque Python qui facilite la création de cartes interactives en utilisant des données géo spatiales. Basée sur la bibliothèque Leaflet.js, Folium permet d'afficher des points, des lignes et des polygones sur des cartes, tout en offrant des options de personnalisation avancées. Dans notre projet, Folium a été utilisé pour visualiser les trajets maritimes des navires sur une carte du monde, permettant aux utilisateurs d'interagir avec les données de manière dynamique.

### **HTML et CSS**

HTML (HyperText Markup Language) est le langage standard utilisé pour créer des pages web. Il permet de structurer le contenu, en utilisant des balises pour définir des éléments comme des titres, des paragraphes, des liens et des images.

CSS (Cascading Style Sheets) est utilisé pour styliser et mettre en forme les documents HTML. Il permet de contrôler l'apparence visuelle des éléments, tels que les couleurs, les polices et les mises en page. Dans notre projet, nous avons utilisé HTML et CSS pour créer une interface utilisateur intuitive, permettant de visualiser les résultats de notre analyse de manière claire et esthétique.

### **SQL**

SQL (Structured Query Language) est un langage standard utilisé pour gérer et manipuler des bases de données relationnelles. Il permet d'effectuer des opérations telles que la création, la lecture, la mise à jour et la suppression de données (CRUD). Dans notre projet, nous avons utilisé SQL pour interroger notre base de données PostgreSQL et récupérer les informations pertinentes sur les trajets des navires. Grâce à des requêtes SQL bien structurées, nous avons pu extraire des données spécifiques et effectuer des analyses approfondies.

## Outils utilisé

### Visual Studio Code

Visual Studio Code (VS Code) est un éditeur de code open-source créé par Microsoft, apprécié pour sa polyvalence et ses fonctionnalités avancées. Il prend en charge de nombreux langages de programmation, dont Python, et offre des outils puissants comme l'auto-complétion, le débogage intégré, et une analyse syntaxique en temps réel. Avec sa large bibliothèque d’extensions, les développeurs peuvent adapter VS Code à leurs besoins. Sa gestion d'environnements virtuels et sa compatibilité avec des outils comme Git en font un choix idéal pour les développeurs, du débutant à l'expert.

### PostgreSQL - PG Admin 4

PostgreSQL est un système de gestion de bases de données relationnelles open-source, réputé pour sa fiabilité et sa puissance. Il permet de gérer efficacement de grandes quantités de données avec des fonctionnalités avancées comme les transactions ACID et les requêtes SQL complexes. PostgreSQL est compatible avec de nombreux langages de programmation et convient aussi bien aux petites applications qu'aux systèmes d'entreprise. Facile à configurer, il bénéficie d'une communauté active qui assure support et mises à jour régulières.

pgAdmin 4 est une interface graphique (GUI) pour PostgreSQL. Il permet aux utilisateurs de gérer et d’administrer leurs bases de données PostgreSQL de manière visuelle, sans avoir besoin d’utiliser la ligne de commande. Avec pgAdmin 4, on peut créer, modifier et supprimer des bases de données, écrire et exécuter des requêtes SQL, visualiser les données, gérer les utilisateurs et les permissions, et surveiller les performances du serveur.

### GitHub Desktop

GitHub Desktop est une application de bureau qui simplifie l'utilisation de Git pour les développeurs, en offrant une interface visuelle intuitive. Elle permet de gérer des dépôts GitHub, de cloner des projets, de suivre les modifications et de gérer les branches sans ligne de commande. L'outil facilite les commits, les merges, et les résolutions de conflits, rendant Git plus accessible aux débutants et pratique pour les utilisateurs expérimentés. GitHub Desktop intègre également des fonctionnalités de synchronisation pour faciliter la collaboration.

# Développement du Projet

## Base de données SQL

Pour notre base de données nous utilisons PostgreSQL, et l’application  « PG Admin 4 » pour avoir une meilleur vision de l’ensemble. Nous devons pour pouvoir effectuer la connexion depuis notre script, créer un utilisateur dans PG Admin 4 et aussi la base de données qui acceuillera les données.

## Ajout des données CSV au SQL

### Création des tables SQL

Notre script Python nommé add\_data\_sql.py nous permet de nous connecter à la base de données et d’interagir avec elle directement depuis celui-ci, nous utilisons la bibliothèque Psycopg2 qui permet la connexion avec la basse de données, et donc avec cette connexion faite, nous pouvons créer des tables depuis notre éditeur de code. (Figure 1)



Figure 1 : Connexion a la base de données et création d'une table SQL

Pour pouvoir avoir des données exploitables, nous devons nous assurer que tout le monde puisse visualiser les mêmes données peut import ou ils se trouvent. C’est pourquoi nous avons dû réaliser des fonctions permettant la conversion des données en des données universel et lisibles par chaque machine. Ce script réalisé en python ce nomme boat\_plugins.py (Figure 2).

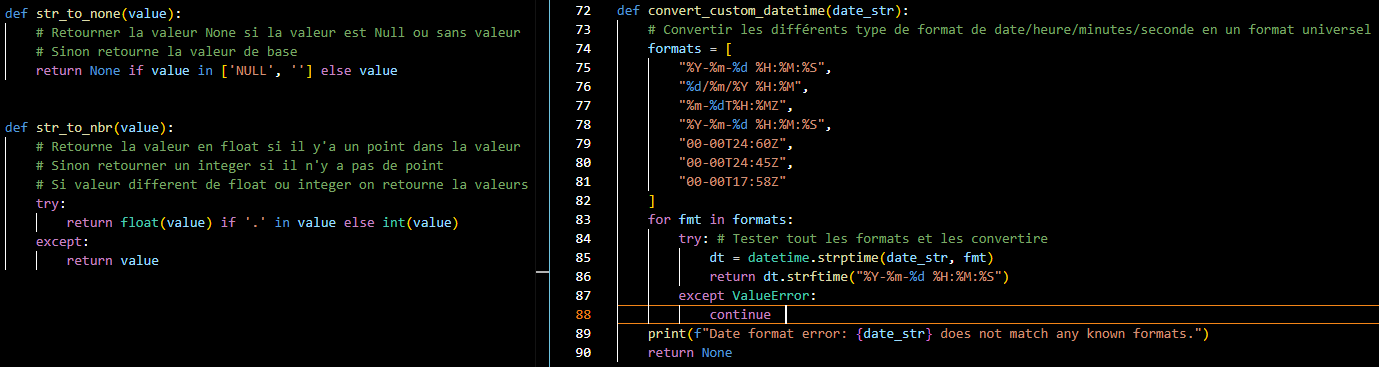
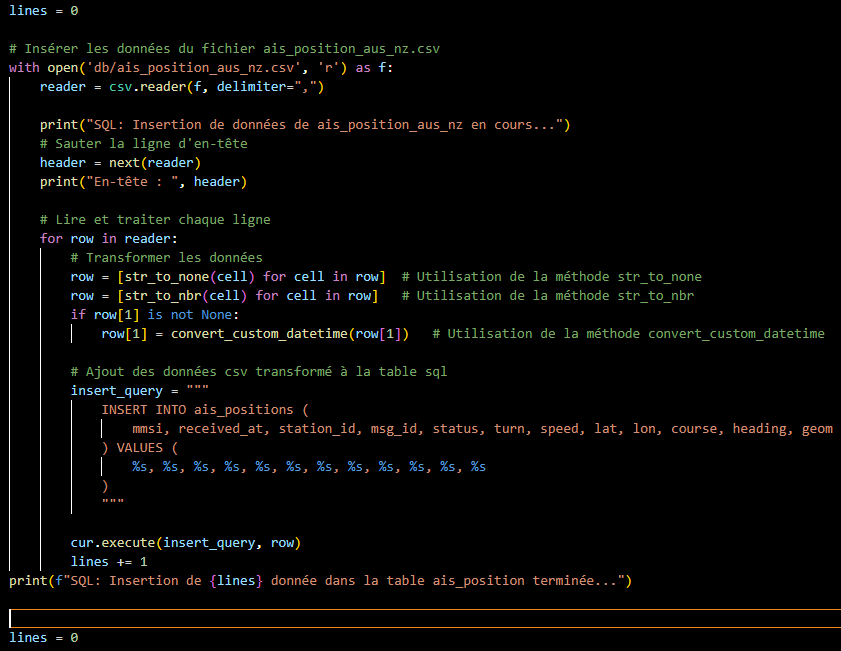


Figure 2 fonctions de conversion

### Insertion des données

Nous avons donc fini la création des tables, maintenant nous devons pouvoir inséré les données CSV, dans notre base de données, pour cela dans notre script add\_data\_sql, nous avons créer une fonction qui permet de faire cela. La fonction ouvre le fichier CSV, et le lit, puis chaque « virgule » ou « point-virgule », nous les utilisons comme des séparateurs. Puis on parcours tout les éléments et on les convertis si besoin est, puis nous insérons les nouvelles données dans la table souhaité.



ANNEXE