

# ESGI - Projet GraphQL / Neo4J / MongoDB

## Les îles Galapagos

Les îles Galápagos appartiennent à l'Équateur, un pays d'Amérique du Sud.

L'archipel est situé à la latitude de l'équateur, à 1000 km à l'ouest des côtes équatoriales. Les Galapagos abritent une quarantaine d'îles volcaniques et la réserve marine des Galápagos, qui constituent un site du patrimoine mondial de l'Unesco.

Les principales îles des Galápagos sont l'île Isabela, l'île San Cristóbal, l'île Santa Cruz et Fernandina...



## Organisation pratique

- Projet en groupe de 3
- Un Github, partagé entre les 3, et avec `rcarlier` à créer au plus tôt
- Soutenance orale avec support (à placer dans le Git en pdf)
- Le rendu final sera également à remettre sur MyGES

A screenshot of a GitHub repository settings page. The left sidebar shows navigation options like 'Access', 'Collaborators' (which is selected and highlighted in blue), 'Moderation options', 'Code and automation', 'Branches', 'Tags', 'Rules', and 'Actions'. The main area is titled 'Public repository' with the sub-instruction 'This repository is public and visible to anyone'. It shows a search bar with the placeholder 'Add people to...' and a text input field containing 'rcarlier'. Below the input field, a search result card for 'Richard Carlier' is displayed, showing a small profile picture and the text 'rcarlier - invite collaborator'. There is also a small 'x' button next to the search bar.

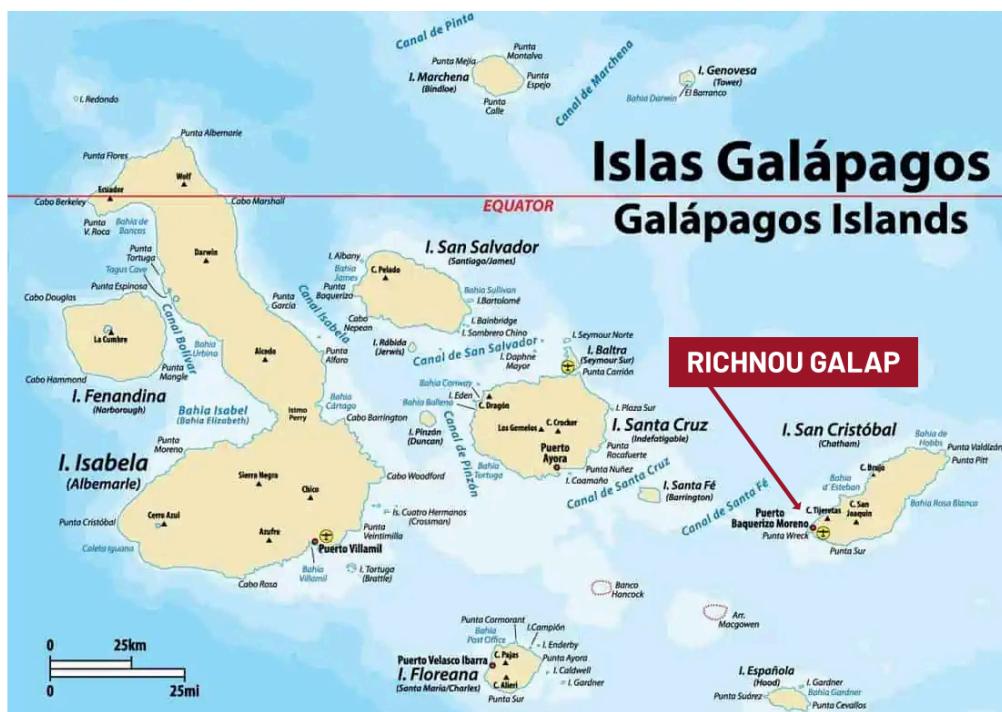
## Contexte

**Richnou Galap** est une société de transport de matériels scientifiques, qui couvre les Galápagos.

Cette société a un entrepot à Puerto Baquerizo Moreno (sur l'île San Cristóbal) et une flotte d'une dizaine d'hydravions. Elle livre ses clients dans des "ports" situés sur les 19 îles principales de l'archipel (la plupart des îles ne possèdent qu'un seul port, mais les plus grandes comme Isabela peuvent en avoir plusieurs), dans des consignes de retrait automatique (lockers) qui sont soit vides, soit pleines (voir plus loin).

Pour simplifier, on peut supposer que la société ne fonctionne qu'avec une dizaine de produits scientifiques à livrer, qu'elle possède en quantité limité - il faut donc gérer les stocks.

Il est important pour la société d'optimiser les livraisons (itinéraire, état des consignes, stock...), et de connaître l'état de la flotte d'hydravions en permanence.



# Objectifs

Votre mission est de monter un système mixant une base de données MongoDB pour une partie des informations, et Neo4J pour une autre, et permettant de répondre à diverses questions, via des requêtes GraphQL. Entre autres:

- Où se trouvent tous les hydravions à un instant T, et leur état (à l'entrepot, au port, en vol, en maintenance, ...)
- On doit livrer x ports avec y clients, quel parcours permet d'optimiser le trajet ( entrepot->port1->port2->...->entrepot ), quelle consommation de carburant cela entraîne ; quel est l'itinéraire optimal pour livrer une série de ports et clients (tenant compte des places dans les lockers) ?
- Quelles "lockers" sont vides/pleines sur les diverses îles
- Quel est l'état des stocks de produits à vendre/livrer
- Quel est l'historique de livraison d'un client
- ...



# Modélisation des données

Il vous est demandé de

- modéliser les données (et répartition Neo/Mongo)
- constituer un jeu de données mixant informations réelles ( coordonnées GPS de port, nom des villes...) et fictives (mais réalistes) et des scripts pour les insérer (il doit être facile de reconstituer la base) - Ne pas hésiter à générer ces données via des scripts pour les automatiser - fournir les scripts.
- un serveur GraphQL pour interroger les données
- une interface légère pour afficher les résultats

Points à intégrer dans la base - non exhaustif:

- hydravions (plusieurs modèles, capacités / consommation de carburant variable)
- clients (des scientifiques qui étudient les volcans ou la faune locale)
- produits (du matériel scientifique)
- commandes, stock
- livraisons (itinéraires, dockers...)
- ...

Points importants à considérer

- les entités, les attributs, les relations
- la répartition entre Neo4J et MongoDB
- langage(s) au choix pour la génération des données, la création du serveur GraphQL, l'interrogation...



## Notes

- on appelle "port" une zone de livraison des hydravions (il peut ne pas y avoir de villes...)
- on suppose un système d'actualisation de la position des hydravions en temps réel
- on calcule la consommation de carburant selon la distance, sans tenir compte des aléas de la météo. De la même façon, on suppose que les hydravions voyagent en ligne droite d'un port à l'autre
- toutes les livraisons se font dans des "caisses" qui ont toutes la même taille. Un client qui passe une commande aura ses produits dans une ou plusieurs caisses, jamais partagées entre 2 clients - un hydravion a une capacité variables de caisses selon le modèle (entre 50 et 150). Chaque locker a une taille correspondant à une caisse. Si un client commande 2 caisses, il faut donc 2 lockers disponibles... On peut envisager des livraisons partielles (une caisse livrée, une autre en attente).



## Compléments

- [Wikipedia](#)
- [Galapagos Information](#)
- [Unesco - Îles Galápagos](#)
- [Récupérer des coordonnées](#)