

# Rapport de SAE

## TITRE

### Fait par :

- Kenza HATEM
- Rémi MILLERAT-GALLOT
- Samir TALBI
- Amokrane AIT TAOUIT
- Ismael LAKRAFLI

## Introduction

Dans un monde toujours plus connecté, la recherche de lieux adaptés à des critères variés — qu'ils soient géographiques, thématiques ou culturels — représente un enjeu de plus en plus crucial. C'est dans ce contexte que notre projet de recherche de lieux multimodaux prend tout son sens.

Ce projet a pour objectif principal de concevoir une application web capable de fournir des informations détaillées, structurées et personnalisées sur des lieux spécifiques, en fonction des préférences et des besoins des utilisateurs. L'ambition est de rendre la navigation et la découverte plus intuitives, tout en établissant des liens enrichissants entre des lieux ou des thèmes variés, qu'il s'agisse de pays, de villes, de sites touristiques, ou encore de catégories thématiques.

## Contexte et objectifs

L'objectif de cette application est de rassembler, structurer et rendre accessibles des données géographiques et thématiques. Contrairement aux moteurs de recherche traditionnels, notre projet repose sur une base de données orientée graphe (Neo4j), qui exploite les liens entre les entités pour produire des résultats enrichis et adaptés à chaque contexte.

### Objectifs principaux :

- **Recherche personnalisée** : Offrir aux utilisateurs la possibilité de rechercher des lieux en fonction de différents critères, tels qu'un thème ou une localisation géographique.
- **Accès simplifié à l'information** : Organiser les données de manière intuitive pour réduire le temps de recherche.

- **Résultats optimisés** : Garantir des réponses rapides et pertinentes, même pour des requêtes complexes.
- **Interface utilisateur attrayante** : Proposer une expérience moderne et adaptée aux attentes des utilisateurs finaux.

### Principales fonctionnalités

L'application offre plusieurs fonctionnalités conçues pour répondre aux besoins variés des utilisateurs :

- **Recherche flexible** : Permet de rechercher des lieux par thème, pays, ville ou lieu précis. Les utilisateurs peuvent saisir un fragment de nom ou sélectionner un thème, comme *histoire* ou *nature*.
- **Résultats enrichis** : Les résultats affichent les relations pertinentes, par exemple, des lieux situés dans un pays ou des thèmes associés à une région.
- **Navigation relationnelle** : Grâce à la base de données graphe, l'utilisateur peut explorer les entités de manière fluide et non linéaire (par exemple, naviguer de *France* à *Paris*, puis aux *Musées de Paris*).
- **Interface intuitive et visuelle** : Les résultats sont présentés de manière hiérarchisée avec des options de filtrage avancées, garantissant une exploration claire et ergonomique.

Ces fonctionnalités, combinées à l'utilisation d'une base de données orientée graphe, permettent une expérience utilisateur unique, alliant performance, fluidité et personnalisation.

## Section 1 : Organisation du Travail d'Équipe

Pour assurer une collaboration fluide et un avancement efficace, nous avons établi des pratiques structurées, tant dans notre communication interne que dans notre gestion collaborative du projet.

### Organisation de la Communication

#### Outils Utilisés

- **Discord** : Communication instantanée pour les échanges quotidiens, les discussions rapides et le partage d'informations critiques.
- **GitLab** : Gestion des bugs, suivi des développements spécifiques et organisation structurée du code.
- **Trello** : Vue d'ensemble des tâches attribuées, gestion des priorités et suivi des délais.

### Réunions Régulières

- **Fréquence** : Une à deux réunions par semaine.
  - **Début de semaine** : Planification des tâches pour les jours à venir.
  - **Fin de semaine (si nécessaire)** : Suivi des progrès, réattribution des tâches non terminées et bilan.
- **Format** : Alternance entre présentiel et distanciel en fonction de la nature des tâches et des disponibilités.

### Gestion Flexible

- **Répartition des tâches** : Formation de groupes (ex. 2-2-1 ou 2-1-1-1) pour équilibrer la charge de travail, en tenant compte des compétences et disponibilités.
- **Résolution collective** : Les tâches complexes sont traitées en équipe, favorisant l'apprentissage mutuel et renforçant l'efficacité.

### Principes de Développement Collaboratifs

#### Méthodologie Agile Inspirée

- **Organisation en sprints hebdomadaires** : Structurer le travail pour maintenir une cadence régulière.
- **Adaptabilité** : Ajustements en temps réel en fonction des imprévus ou des retours lors des revues hebdomadaires.

#### Collaboration via GitLab

- **Branches dédiées** : Une branche par fonctionnalité ou tâche pour éviter les conflits de code.
- **Merge requests** : Revue de code obligatoire pour garantir la qualité et partager les connaissances techniques.

#### Pair Programming et Soutien Mutuel

- **Tâches critiques ou complexes** : Abordées en pair programming pour échanger des idées et résoudre efficacement les problèmes.
- **Apprentissage continu** : Les membres expérimentés accompagnent les autres, favorisant une montée en compétences collective.

## Résolution Collaborative

- **Partage rapide** : Problèmes bloquants partagés sur Discord pour obtenir des solutions rapidement.
- **Sessions collectives** : Organisation de sessions d'équipe pour résoudre les problèmes complexes, assurant ainsi une efficacité maximale.

## Tests et Validation

- **Double validation** : Chaque fonctionnalité est testée par son développeur et un autre membre de l'équipe.
- **Scénarios réels et limites** : Tests approfondis pour garantir la robustesse et la qualité de l'application.

En combinant une communication structurée et des principes collaboratifs solides, notre équipe a construit un environnement de travail efficace et convivial, permettant de livrer un produit final de qualité.

## Section 2 : Ingénierie Logicielle

### 1. Présentation Générale de l'Application

L'application a été conçue pour faciliter la recherche de lieux multimodale selon différents critères (thèmes, noms, associations géographiques, etc.). Elle propose une interface conviviale et réactive, permettant aux utilisateurs d'explorer efficacement les liens entre divers éléments, tels que les pays, villes, lieux et thèmes associés.

### Principales Situations d'Utilisation

- **Recherche par nom** : L'utilisateur entre un mot-clé (fragment d'un nom), et l'application retourne tous les résultats correspondants (pays, villes ou lieux associés).
- **Exploration contextuelle** : Une recherche thématique affiche les lieux liés au thème sans inclure directement le thème lui-même.
- **Navigation intuitive** : Un système de filtres interactifs permet de naviguer facilement entre les relations des entités.

### 2. Structure et Architecture de l'Application

**Backend Langage utilisé : TypeScript avec Deno**

- Pourquoi TypeScript ?

TypeScript garantit un typage fort, réduisant ainsi les erreurs de développement.

Il offre également une meilleure lisibilité du code et facilite sa maintenance à long terme.

- **Pourquoi Deno ?**

Deno présente plusieurs avantages techniques qui en font un choix pertinent pour notre application :

- **Sécurité accrue** : Permissions explicites par défaut. Par exemple, sans spécification explicite, le runtime n'a pas accès au système de fichiers, au réseau ou à d'autres opérations sensibles.
- **Gestion simplifiée des dépendances** : Élimine le besoin de fichiers comme `package.json` et l'utilisation encombrante du dossier `node_modules`.
- **Support natif de TypeScript** : Permet de réduire la configuration initiale, accélérant ainsi le démarrage du projet.
- **Performance et légèreté** : Basé sur le moteur V8, Deno est rapide et léger tout en fournissant des bibliothèques modernes adaptées à notre workflow.
- **Approche minimaliste et sécurisée** : En supprimant les éléments superflus et en instaurant une architecture sécurisée dès le départ, Deno simplifie le processus de développement.

En résumé, l'utilisation conjointe de TypeScript et Deno garantit une architecture backend moderne, robuste et sécurisée, tout en offrant une expérience de développement fluide.

Lien vers le gitlab : [Gitlab](#)

VIDEO DE PRESENTATION

## Section 3 : Optimisation

### Requêtes vers la base de données

Base de données utilisée : Neo4j

Langage : Cypher

Cypher est un langage optimisé et performant pour interroger Neo4j, particulièrement adapté pour naviguer dans des graphes complexes.

### Optimisations mises en place

#### 1. Pagination pour les résultats

- **Problème initial** : Lors du chargement de tous les nœuds sans limitation, l'application rencontrait des problèmes de performance.

- **Solution :**
  - Afficher 60 résultats par page et implémenter un système de pagination.
  - Cela réduit la charge sur le serveur et améliore l'expérience utilisateur.

## 2. Optimisation du chargement des images

- **Problème initial :** Les images associées aux résultats étaient stockées en URL, souvent issues de Wikipedia. Cela entraînait des temps de chargement longs, provoquant des ralentissements, même avec un petit nombre de résultats.
- **Solutions mises en place :**
  - **Suppression des images sur la page d'accueil :** Les images dynamiques ont été remplacées par des images par défaut pour chaque type de label (continent, ville, lieu, etc.).
  - **Préchargement des images en cache :** Lors du lancement d'une recherche, toutes les images associées aux résultats sont préchargées et stockées en cache. Ainsi, lorsqu'une suggestion est sélectionnée, l'image correspondante s'affiche instantanément, améliorant l'expérience utilisateur.

## 3. Ajout du debouncing

- **Problème initial :** Lors de la saisie dans la barre de recherche, chaque lettre était immédiatement traitée, entraînant de nombreux appels réseau inutiles.
- **Solution :** Mise en place d'un système de debouncing pour réduire le nombre d'appels réseau superflus.
  - **Avantage :** Amélioration des performances globales.
  - **Inconvénient :** Introduit un petit délai (quelques millisecondes) avant l'affichage des résultats, ce qui peut être perceptible pour l'utilisateur.

Ces optimisations permettent d'offrir une navigation fluide et réactive, même avec un grand volume de données.

## Mise en place des métriques de performance dans l'application

Pour surveiller et visualiser les métriques de performance de l'application, nous avons implémenté une solution complète permettant de suivre les performances des API, notamment les temps de réponse et le nombre d'éléments retournés.

Les métriques sont enregistrées dans la base de données Neo4j et récupérées via un endpoint dédié `/metrics`. Ces données sont affichées sur le frontend dans un graphique interactif, créé avec React et Chart.js, permettant aux utilisateurs d’analyser les tendances au fil du temps. Des fonctionnalités telles que l’agrégation (par exemple, moyennes sur 10 secondes, 1 minute ou 5 minutes) et la pagination ont été mises en œuvre pour gérer efficacement de grands ensembles de données. Des informations supplémentaires, notamment les temps de réponse maximum, minimum et moyen pour la page actuelle, sont affichées sous le graphique.

Grâce à ce système, nous avons pu observer une forte réduction du temps de réponse de chaque requête après la mise en place des optimisations évoquées précédemment, comme la pagination, l’optimisation du chargement des images, et enfin du debouncing.

### **Bonus : Récupération des données depuis Wikidata**

Pour récupérer les données que nous avons importées dans notre base de données Neo4J, nous avons utilisé l’interface de requête `query.wikidata.org` qui fonctionne via des requêtes SPARQL. Une limitation majeure était le timeout après 60 secondes : nous avons un peu trop de données à récupérer pour que ce soit gênant, mais pas assez pour choisir de plutôt faire un dump et traiter les données nous-même.

En particulier, plus on récupère d’informations par item, et plus on trie ces informations, plus la requête est longue. Une grande partie de la difficulté était donc de ne pas trop récupérer d’informations, ni trop peu, pour que derrière ça ne soit pas trop difficile à nettoyer avec Python, mais pas non plus trop long par rapport au timeout, tout en gardant suffisamment de données pour s’en servir dans l’application.

## **Section 4 : Aspects Communicationnels**

### **Identité Visuelle et Branding**

#### **Nom et Concept**

Le choix du nom *Destina* est simple, fluide et chargé de sens. Inspiré du mot “destination”, il évoque immédiatement l’idée de voyage, de découverte et d’exploration. Ce nom court et percutant reflète l’accessibilité de l’application, tout en restant universel et mémorable.

Le slogan *Cherchez. Trouvez. Partez* renforce cette identité en résumant parfaitement l’objectif de l’application : permettre aux utilisateurs de trouver leur prochaine destination en quelques clics. La structure en trois verbes crée un rythme dynamique, capturant l’essence du parcours utilisateur.

#### **Palette de Couleurs**

La palette choisie pour *Destina* mélange des teintes apaisantes et professionnelles, allant du bleu clair au bleu marine. Ces couleurs ne sont pas seulement

esthétiques ; elles traduisent des valeurs fondamentales :

- **#007FA9 (couleur primaire)** : Ce bleu vif inspire la confiance, la sérénité et la modernité. Il est utilisé pour les éléments interactifs clés comme les boutons ou les liens.
- **#CEDDDE, #9CD2D5, #82CFD8** : Des nuances de bleu et de turquoise qui évoquent l'eau, la nature et l'évasion. Ces teintes sont idéales pour des arrière-plans ou des sections apaisantes.
- **#00334A** : Une couleur sombre pour les titres et les éléments structurants, offrant une excellente lisibilité et une touche d'élégance.

Ensemble, ces couleurs créent une identité visuelle cohérente, moderne et adaptée à une application orientée voyage et exploration.

### Typographie

Deux familles de polices ont été sélectionnées pour garantir lisibilité et modernité :

- **Poppins et Montserrat (polices principales)** :  
Ces polices géométriques et minimalistes sont parfaitement adaptées à une application moderne. Elles offrent une excellente lisibilité, même sur de petits écrans, tout en conservant une esthétique épurée et professionnelle.
  - *Montserrat* est utilisée pour les titres, renforçant l'impression de clarté et d'organisation.
  - *Poppins*, plus arrondie et fluide, s'adapte aux boutons, menus et sections interactives.
- **Open Sans et Roboto Medium (polices secondaires)** :  
Ces polices secondaires garantissent une lecture confortable pour les descriptions longues ou les textes explicatifs. Leur simplicité et leur lisibilité les rendent idéales pour les paragraphes informatifs ou les détails des destinations.

## Stratégie de Communication et Branding

### Campagne de lancement

Pour présenter *Destina*, une vidéo courte et impactante sera créée. Cette vidéo, d'une durée de 30 à 60 secondes, mettra en avant les fonctionnalités clés de l'application, telles que la recherche rapide, les suggestions thématiques et l'interface intuitive.

### Affiche de Présentation

Pour illustrer visuellement l'identité de *Destina*, nous avons créé une **affiche** combinant éléments textuels et visuels :



- **Mise en avant** du logo et des couleurs clés (vert grisé et blanc), rappelant le thème du voyage et de la détente.
- **Forme arrondie et découpée** pour accueillir la photographie, évoquant l'idée de découverte et de lieux uniques.
- **Liste de mots-clés** ("NATURE", "VACANCES", "DÉCOUVERTES", etc.) reflétant la diversité des destinations proposées.

Cette composition permet d'affirmer le **positionnement** de *Destina* comme une solution moderne, à la fois **sobre** et **évocatrice**, axée sur la découverte et l'évasion.

### Lettre de Bienvenue

Nous avons également conçu une **lettre de bienvenue** à l'attention des voyageurs, leur souhaitant la bienvenue sur *Destina*.

- **Tonalité chaleureuse** : Le texte s'adresse directement à l'utilisateur ("Chers Voyageurs...") pour instaurer une relation de proximité.
- **Couleurs harmonieuses** : Le bandeau supérieur bleu marine, surplombé du logo, donne une dimension institutionnelle et rassurante.
- **Présentation claire** : Le message principal reste concis ; il souligne la simplicité d'usage de l'application et invite l'utilisateur à explorer les nombreuses fonctionnalités.

En quelques phrases, cette lettre installe une **atmosphère conviviale** et rappelle la **promesse** de l'application : "Cherchez. Trouvez. Partez !"

IMAGE 1 IMAGE 2

### Positionnement Marketing

*Destina* se distingue par son approche gratuite et accessible à tous. En misant sur la simplicité d'utilisation et la richesse des résultats, l'application cible un large public :

- **Grand public** : Pour des besoins quotidiens, comme la recherche de lieux pour les vacances ou les week-ends.
- **Curieux et exploreurs** : Ceux qui souhaitent découvrir des destinations moins connues ou thématiques. Secti

## Section 5 : Rapport Utilisateur (camarade de classe)

Dans le cadre de mes tests utilisateur, j'ai eu l'opportunité d'évaluer l'application **Destina**, qui propose une expérience enrichissante pour découvrir

des destinations inédites. Voici mon retour détaillé sur cette application :

### Points Positifs

- **Résultats rapides et personnalisés** : L'application offre des suggestions adaptées aux besoins et intérêts de chaque utilisateur, ce qui garantit une expérience sur mesure.
- **Découverte de lieux exclusifs** : Destina met en avant des endroits peu connus et rarement répertoriés sur d'autres sites ou plateformes, ce qui ajoute une touche d'originalité.
- **Interface intuitive et ergonomique** : La navigation est fluide, claire et agréable grâce à un design bien pensé. L'interface est conviviale, même pour les utilisateurs novices.
- **Images indicatives** : Chaque lieu est accompagné de visuels explicites, permettant de mieux visualiser les destinations proposées.
- **Gestion simplifiée des favoris** : Les utilisateurs peuvent facilement ajouter des lieux à leur liste de favoris et les gérer en quelques clics, ce qui est pratique pour planifier des voyages.
- **Suggestions de thèmes innovants** : L'application propose des thèmes uniques, tels que des lieux historiques, mystérieux ou naturels, incitant à sortir des sentiers battus.

### Points Négatifs

- **Descriptions parfois limitées** : Certaines destinations manquent de détails approfondis sur leur histoire, leur localisation ou les activités disponibles, ce qui pourrait être frustrant pour les utilisateurs curieux.
- **Absence d'images pour certains lieux** : Certains endroits sont dépourvus de visuels, ce qui oblige l'utilisateur à effectuer des recherches externes pour se faire une idée du lieu.

### Recommandations

Pour améliorer encore l'expérience utilisateur, il serait intéressant :

1. D'enrichir les descriptions des lieux avec plus d'informations (histoire, anecdotes, activités possibles, etc.).
2. D'ajouter des images pour toutes les destinations, ou au moins des illustrations par défaut pour éviter les zones "blanches".

3. D'inclure une option permettant aux utilisateurs de laisser des avis ou des photos sur les lieux qu'ils ont visités.

## Conclusion

En conclusion, l'application Destina est un outil puissant et innovant pour les voyageurs en quête de nouvelles expériences. Avec quelques ajustements, elle pourrait devenir une référence incontournable dans son domaine.

## Section 6 Bonus : Licence

### Choix de la licence : EUPL (European Union Public License)

Dans le cadre de cette SAE, nous avons décidé d'adopter la **European Union Public License (EUPL)** pour notre application de recherche de destinations touristiques. Cette licence s'applique à l'ensemble des composants que nous développons : la base de données Neo4J enrichie à partir de Wikidata, le backend en Deno, et le frontend en React. Voici les raisons principales qui motivent ce choix :

#### 1. Compatibilité avec les données et l'écosystème open source

Notre projet utilise des données issues de **Wikidata**, une ressource sous licence **CC0**. Ces données sont explicitement libres de droits, ce qui permet leur réutilisation dans des projets sous toute licence open source. L'EUPL est particulièrement adaptée, car elle est compatible avec un large éventail d'autres licences open source, notamment la **GPL v3**, **AGPL**, et **MPL 2.0**. Cette compatibilité garantit que notre application pourra être intégrée ou utilisée dans des projets tiers respectant également une philosophie de partage.

#### 2. Une licence adaptée aux logiciels

L'EUPL est une licence conçue pour les logiciels et adaptée aux environnements techniques modernes. Elle couvre l'ensemble des composants de notre projet, qu'il s'agisse du code source du backend, du frontend, ou des requêtes et scripts liés à Neo4J. Contrairement à des licences comme la **CC BY-SA**, qui ciblent les œuvres créatives, l'EUPL est spécifiquement pensée pour les projets logiciels et intègre des clauses pertinentes, comme la nécessité de distribuer le code source des modifications.

#### 3. Obligation de partage des contributions

L'EUPL impose un **copyleft fort**, ce qui signifie que toute modification ou redistribution de notre application devra être effectuée sous la même licence ou une licence compatible (par exemple, GPL v3). Cela garantit que les utilisateurs de notre projet, y compris les futures équipes universitaires ou des développeurs externes, contribueront à l'écosystème open source en redistribuant leurs améliorations dans les mêmes conditions.

#### 4. Attribution des auteurs

Contrairement à certaines licences open source comme la **GPL v3**, l'EURL exige explicitement une **attribution des auteurs**. Cela garantit que le travail de notre équipe sera reconnu dans les versions dérivées ou redistribuées de l'application, qu'il s'agisse de modifications mineures ou de réutilisation dans des projets plus larges. Cette exigence est essentielle dans un cadre universitaire, où la reconnaissance des contributions est un enjeu important.

#### 5. Adaptation au cadre juridique européen

L'EURL est spécifiquement rédigée pour s'aligner sur le droit européen, ce qui offre une sécurité juridique accrue, notamment pour un projet académique réalisé dans l'Union européenne. Cette spécificité est un avantage par rapport à d'autres licences globales comme l'**Apache 2.0**, qui peuvent manquer de clarté sur certains points lorsqu'elles sont appliquées dans des juridictions européennes.

#### Conclusion

Le choix de l'EURL s'impose comme la solution la plus appropriée à notre projet pour plusieurs raisons : sa compatibilité avec les données et licences open source, son copyleft fort qui encourage le partage, son exigence d'attribution qui valorise notre travail, et son adaptation au cadre juridique européen. Grâce à cette licence, nous contribuons à promouvoir une culture d'innovation ouverte, tout en assurant une reconnaissance durable de nos contributions au projet.