

UNIVERSITÉ DE MONTPELLIER  
M1 ARCHITECTURE DU GÉNIES LOGICIEL ET DU WEB

## Conduite de projet

RAPPORT DE PROJET  
CONDUITE DE PROJET — HMIN204

**Étudiants :**

Denis BEAUGET Hayaat HEBIRET

**Année :** 2020 – 2021

**Encadrant :**

Eric BOURREAU



UNIVERSITÉ  
DE MONTPELLIER



# Table des matières

<b>1</b>	<b>Data'Clear</b>	<b>2</b>
1.1	Résumé . . . . .	2
1.2	Objectifs . . . . .	2
<b>2</b>	<b>Planning</b>	<b>4</b>
2.1	Planning prévisionnel . . . . .	4
2.2	Planning final . . . . .	5
<b>3</b>	<b>Données</b>	<b>8</b>
3.1	Données quantitatives . . . . .	8
3.1.1	Nombre de commit . . . . .	9
3.1.2	Nombre de classes . . . . .	10
3.1.3	Nombre de lignes de code . . . . .	10
3.1.4	Nombre d'heures . . . . .	12
3.1.5	Issues . . . . .	12
3.1.6	Features en cours . . . . .	12
3.2	Données qualitatives . . . . .	12
3.2.1	Comparatif des plannings . . . . .	12
3.2.2	Explication des différences . . . . .	13
<b>4</b>	<b>Conclusion</b>	<b>14</b>
	<b>Annexes</b>	<b>15</b>

# Chapitre 1

## Data'Clear

### 1.1 Résumé

Data'Clear est un projet de la science des données dont l'objectif principal est de répondre à des problématiques actuelles liées à ce domaine. Cette idée de projet est née pendant les cours du premier semestre comme "HMIN122M - Entrepôts de Données et Big-Data", nous voulions répondre à des questions comme : Pourquoi l'accessibilité aux données et à leurs études est si compliquée pour le grand public ? Pourquoi les solutions de stockage de données sont si complexes ? A travers un cas d'étude : Les données du naufrage du Titanic, nous sommes parties de sources hétérogènes de données (les 3 classes de passagers ayant embarqué dans le Titanic, dans des formats de données différents) et nous avons effectué tout le "parcours" de ces données, de la réflexion pour une solution de stockage pérenne (SGBD/NoSQL) et un cadre dynamique à travers un modèle ontologique (Mappings, UML) à l'extraction de nos données transformées (via plusieurs méthodes : réseau, fichiers) et à l'étude de nos résultats via Python.

### 1.2 Objectifs

- Stocker les données dans des bases adaptées
- Fournir un schéma global pour répondre à l'hétérogénéité des données
- Extraire ces données mapper à partir de notre schéma
- Analyser nos données pour mettre en place un modèle de prédiction pour la survie des passagers

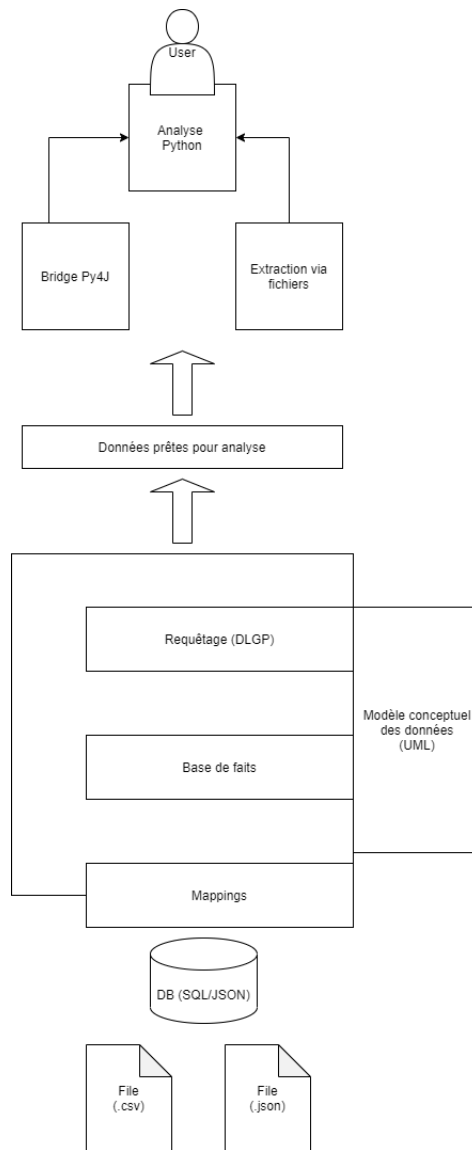


FIGURE 1.1 – Schéma de notre application

# Chapitre 2

## Planning

Ces 2 plannings sont disponibles via le lien suivant : <https://github.com/Beauget/Data-Clear/tree/main/Ressources/ConduiteProjet>

### 2.1 Planning prévisionnel

Ci-dessous le planning prévisionnel de notre projet et les tâches associées.

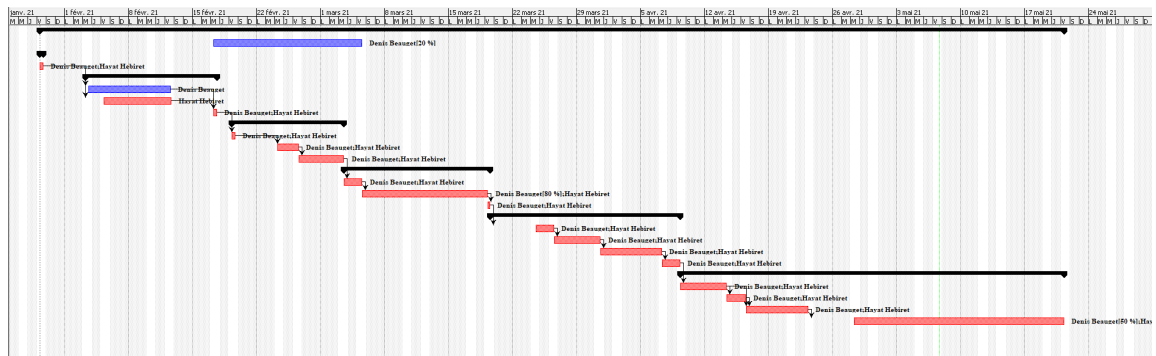


FIGURE 2.1 – Planning prévisionnel


























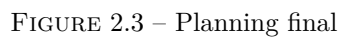
Travail		Nom	Durée	Début
409 heures		 <b>Data'Clear</b>	32,156 jou...	29/01/21 08:00
9 heures		Planning Prévisionnel	5,625 jours	16/02/21 08:00
16 heures		 <b>Etudes bibliographique</b>	1 jour	29/01/21 08:00
16 heures		Etudes articles (EP)	1 jour	29/01/21 08:00
64 heures		 <b>Chargement des données</b>	5 jours	03/02/21 08:00
24 heures		BDD NoSQL	3 jours	03/02/21 08:00
24 heures		BDD SQL	3 jours	03/02/21 08:00
16 heures		Test + Rapport MAJ	1 jour	17/02/21 08:00
60 heures		 <b>Définir le modèle ontol</b>	3,75 jours?	19/02/21 08:00
16 heures		Etudes détaillés	1 jour?	19/02/21 08:00
30 heures		Définir le modèle	1,875 jours	24/02/21 08:00
14 heures		Ecrire les règles	0,875 jours	26/02/21 16:00
68 heures		 <b>Intégrer les données</b>	4,906 jours?	03/03/21 15:00
16 heures		Implémentation règles	1 jour?	03/03/21 15:00
42 heures		Mappings via le modèle	3,281 jours?	05/03/21 15:00
10 heures		Test + Rapport MAJ	0,625 jours?	19/03/21 08:15
72 heures		 <b>Graal</b>	5,5 jours?	19/03/21 14:15
16 heures		Etudes préalables	2 jours?	19/03/21 14:15
20 heures		Intégrer le modèle à Graal	1,25 jours?	26/03/21 14:15
20 heures		Appliquer le mappings sur	1,25 jours?	31/03/21 16:15
16 heures		Test + Rapport MAJ	1 jour?	07/04/21 09:15
120 heures		 <b>Apprentissage et prédi</b>	12 jours?	09/04/21 09:15
20 heures		Mise en place d'apprentis	1,25 jours?	09/04/21 09:15
20 heures		Résultat des prédictions	1,25 jours?	14/04/21 11:15
24 heures		Visualisation	1,5 jours	16/04/21 14:15
56 heures		Rapport + Bilan	8 jours?	23/04/21 09:15

FIGURE 2.2 – Tâche prévisionnel

## 2.2 Planning final

Ci-dessous le planning final de notre projet et les tâches associées.































































Travail		Nom	Durée	Début
395,2 heures	 	 <b>Data'Clear</b>	34 jours?	<b>29/01/21 08:00</b>
3,2 heures	 	Planning Prévisionnel	2 jours	17/02/21 08:00
16 heures	 	 <b>Etudes bibliographique</b>	1 jour	<b>29/01/21 08:00</b>
16 heures	 	Etudes articles (EP)	1 jour	29/01/21 08:00
64 heures	 	 <b>Chargement des données</b>	4,125 jours	<b>03/02/21 16:00</b>
24 heures	 	BDD NoSQL	3 jours	03/02/21 16:00
24 heures	 	BDD SQL	3 jours	05/02/21 08:00
16 heures	 	Test + Rapport MAJ	1 jour	17/02/21 08:00
60 heures	 	 <b>Définir le modèle ontologique</b>	3,75 jours?	<b>19/02/21 08:00</b>
16 heures	 	Etudes détaillées	1 jour?	19/02/21 08:00
30 heures	 	Définir le modèle	1,875 jours	24/02/21 08:00
14 heures	 	Ecrire les règles	0,875 jours	26/02/21 16:00
68 heures	 	 <b>Intégrer les données</b>	4,906 jours?	<b>03/03/21 15:00</b>
16 heures	 	Implémentation règles	1 jour?	03/03/21 15:00
42 heures	 	Mappings via le modèle	3,281 jours?	05/03/21 15:00
10 heures	 	Test + Rapport MAJ	0,625 jours?	19/03/21 08:15
72 heures	 	 <b>Graal</b>	4,5 jours?	<b>24/03/21 14:15</b>
16 heures	 	Etudes préalables	2 jours?	24/03/21 14:15
20 heures	 	Intégrer le modèle à Graal	1,25 jours?	26/03/21 14:15
20 heures	 	Appliquer le mappings sur	1,25 jours?	31/03/21 16:15
16 heures	 	Test + Rapport MAJ	1 jour?	07/04/21 09:15
112 heures	 	 <b>Apprentissage et prédiction</b>	13 jours?	<b>14/04/21 08:00</b>
20 heures	 	Mise en place d'apprentissage	1,25 jours?	14/04/21 08:00
20 heures	 	Résultat des prédictions	1,25 jours?	16/04/21 10:00
24 heures	 	Visualisation	1,5 jours	21/04/21 12:00
48 heures	 	Rapport + Bilan	7 jours?	05/05/21 08:00

FIGURE 2.4 – Tâche final



## Chapitre 3

# Données

### 3.1 Données quantitatives

Lien vers le git du projet : <https://github.com/Beauget/Data-Clear>

### 3.1.1 Nombre de commit

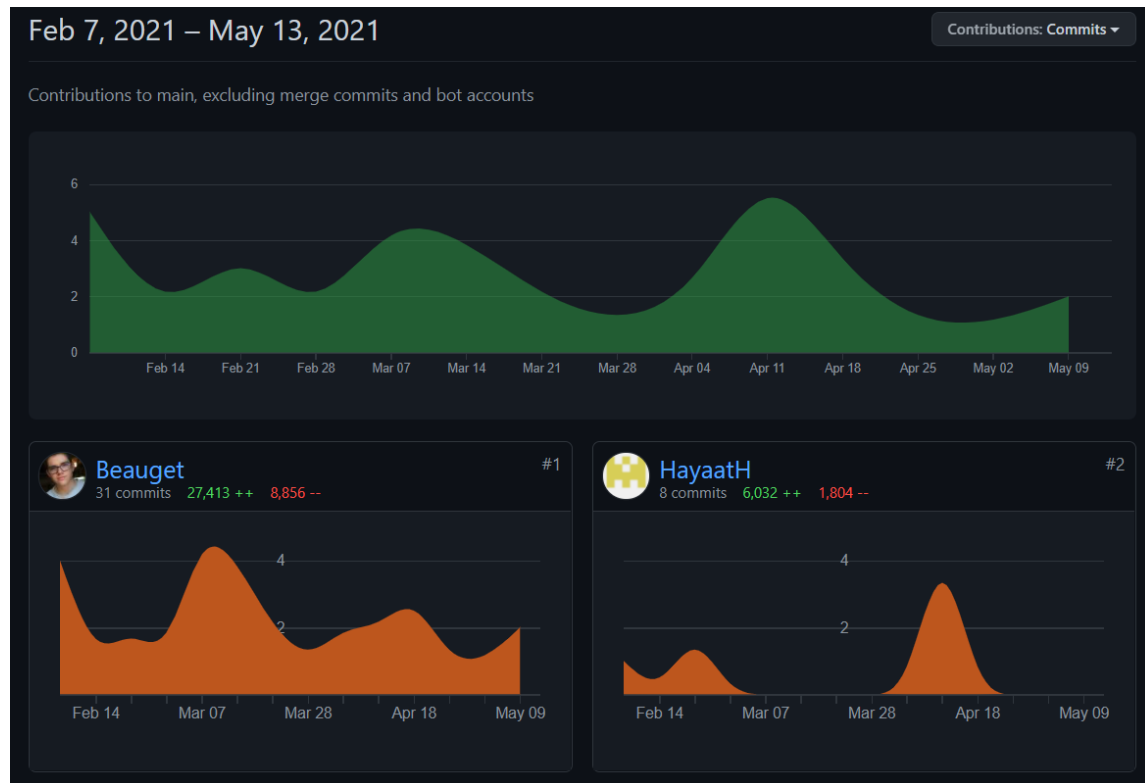


FIGURE 3.1 – Résumé données du Git

Au moment où nous écrivons ce rapport nous avons fait 39 commits sur le Git du projet, nous avons privilégiés des commits réguliers, parfois léger, mais en gardant l'objectif de suivre notre planning.

On peut aussi remarquer que Denis push beaucoup plus souvent qu'Hayaat, cela est lié à notre façon de travailler ce projet. Nous avons 2 jours par semaine ensemble consacré au TER, nous utilisons l'outil CodeTogether<sup>1</sup> qui permet de coder "en même temps" sur le même projet, à la fin des sessions (ou dans les jours qui suivent après relecture par exemple) Denis envoyait la "brique" de la semaine ou de la dernière session avant la réunion avec les encadrants. Cela permettait une meilleure communication au sein du groupe et ainsi, chacun comprenait la feature en cours.

1. CodeTogether <https://www.codetogether.com/>

### 3.1.2 Nombre de classes

Sur la figure qui suit, on observe toutes les classes créées pendant le projet, nous avons décidé de mettre sur le Git que les fichiers de classe bruts et pas l'intégralité de l'arborescence du projet eclipse (pour faciliter la compréhension). Pour la partie Java du projet (avant la phase d'analyse) nous avons donc 10 classes.

CabineRelation.java	add SQLtoMapping class + sqlite 34 instr()	last month
ClasseRelation.java	add SQLtoMapping class + sqlite 34 instr()	last month
EmbarqueRelation.java	add SQLtoMapping class + sqlite 34 instr()	last month
GetDatabase.java	add mongoDb mapping	23 days ago
MongoDBMappingEvaluator.java	end python method	5 days ago
PassagerMappingExemple.java	add SQLtoMapping class + sqlite 34 instr()	last month
PassagerRelation.java	add SQLtoMapping class + sqlite 34 instr()	last month
SQLMappingEvaluator.java	start python + export query	last month
VoyageTitanicRelation.java	add SQLtoMapping class + sqlite 34 instr()	last month
mainExemple.java	end python method	5 days ago

FIGURE 3.2 – Fichiers de classes Graal

..		
.ipynb_checkpoints	end python method	10 days ago
analyze.ipynb	maj analyze	9 days ago
back.csv	add query DLGP python	19 days ago
check.csv	end python method	10 days ago
data.csv	end python method	10 days ago
desktop.ini	start python + export query	last month
py4jDLGP.py	Py4J bridge + external evaluate()	last month
titanic.preprocessing.csv	end python method	10 days ago

FIGURE 3.3 – Fichiers d'analyse Python

### 3.1.3 Nombre de lignes de code

Nous avons travaillé sur plusieurs points au cours du projet, nos lignes de code se découpent en 3 points.

#### - Compréhension du mapping via SQLite

```
C:\Users\beaug\Desktop\back\Data-Clear\Mapping\MappingSQLITE\Fonctions java\src\csvToSQLITE (main -> origin)
λ ls | wc -l SQLiteJDBC.java
332 SQLiteJDBC.java
```

FIGURE 3.4 – Mapping csv =&gt; SQLite

```
C:\Users\beaug\Desktop\back\Data-Clear\Mapping\MappingSQLITE\Fonctions java\src\SQLITetoSQLITE (main -> origin)
λ ls | wc -l SQLITetoSQLITE.java
509 SQLITetoSQLITE.java
```

FIGURE 3.5 – Mapping SQLite =&gt; SQLite (avec notre modèle UML)

### - Mapping des données issu de nos bases avec l'outil Graal<sup>2</sup>

```
C:\Users\beaug\Desktop\back\Data-Clear\Gaal\FichierGaal (main -> origin)
λ ls | wc -l CabineRelation.java ClasseRelation.java EmbarqueRelation.java VoyageTitanicRelation.java PassengerRelation.java PassengerMappingExemple.java mainExemple.java GetDatabase.java SQLMappingEvaluator.java MongoDBMappingEvaluator.java
63 CabineRelation.java
51 ClasseRelation.java
52 EmbarqueRelation.java
66 VoyageTitanicRelation.java
53 PassengerRelation.java
278 PassengerMappingExemple.java
393 mainExemple.java
39 GetDatabase.java
66 SQLMappingEvaluator.java
69 MongoDBMappingEvaluator.java
1130 total
C:\Users\beaug\Desktop\back\Data-Clear\Gaal\FichierGaal (main -> origin)
λ |
```

FIGURE 3.6 – Classe de mapping via Graal

### - Analyse des données issue de nos mappings via Python/Jupyter

```
C:\Users\beaug\Desktop\back\Data-Clear\Python (main -> origin)
λ ls | wc -l analyze.ipynb
1921 analyze.ipynb
```

FIGURE 3.7 – Fichier d'analyse Python

2. Graal : <https://graphik-team.github.io/graal/>

### 3.1.4 Nombre d'heures

C'est assez difficile d'estimer le nombre d'heures total pour le projet mais, nous pouvons en déduire une partie.

- **9 réunions** d'approximativement 1h30 chacune soit environ **13h30** de réunion
- **2 jours/semaine** consacré uniquement au projet environ **12 heures** par semaine sur environ 16 semaines soit **192 heures/personne**
- Notre lecture des articles et la documentation estimé à **20 heures**
- Les tests et les mises à jour du rapport régulières environ **90 heures**

On arrive à **315 heures/personne** nous ne pouvons pas déterminer exactement le temps passé hors des jours de travail (correction mineur après discussion avec les encadrants, préparation des présentations...).

### 3.1.5 Issues

Un problème rencontré récemment auquel nous avons dû trouver une solution c'est la démonstration de notre travail. En effet, comme indiqué nos données sont stockées dans des bases de données SQL et NoSQL à la base pour nos extractions. Il était difficile de créer des accès distants et de l'expliquer, nous avons donc fait une vidéo qui reprend les éléments du projet en suivant la chronologie de notre travail.

Lien de la vidéo : <https://youtu.be/piaip86IgtQ>

### 3.1.6 Features en cours

Une feature que nous aimerions développer pendant l'été (une sorte d'extension du projet) c'est d'adapter notre modèle à un autre cas d'étude (nous pensions à des données issues de la restauration par exemple).

## 3.2 Données qualitatives

### 3.2.1 Comparatif des plannings

Les plannings sont assez ressemblants à première vue, mais il y a quand même quelques différences mineur à souligner.

Première différence, le temps consacré au planning prévisionnel a été divisés de moitié (même plus).

Seconde différence qui apparaît sur la fin du projet, entre l'avant-dernière et la dernière tâche, il y a une "pause" entre le bilan du projet et la finalisation du rapport et de la présentation et la fin de la visualisation. Le nombre d'heures consacré à cette tâche final a également été réduit.

### 3.2.2 Explication des différences

Le temps consacré au planning prévisionnel s'explique en 2 points. Le premier, nous nous attendions que d'autre élève rejoigne notre projet et nous avions prévu de pouvoir accueillir d'autres membres si ils n'avaient pas trouvé de groupe, ce qui n'a pas été le cas. De ce fait nous devons faire un planning pour 2 personnes au final. Le second point, c'est que les Travaux Pratiques de l'UE Conduite de Projet nous ont aidé pour réaliser notre planning bien plus facilement. Ces 2 arguments explique la réduction du temps accordé à cette tâche.

Pour la seconde différence, cela est plutôt dû à un imprévu. Nous avons eu l'annonce des examens et de certains oraux prévu pour début mai ce qui nous aurait mis en retard sur notre projet. Nous avons donc décidé de décaler la fin du planning pour nous laisser le temps de réviser dans l'objectif d'être concentré uniquement sur notre projet par la suite.

## Chapitre 4

# Conclusion

Pour ce TER c'est la première fois que nous étions que 2 pour un projet. Cela peut sembler plus facile à gérer, d'un point de vue planning et répartition du travail, car il y a moins de personnes, mais il n'y a pas moins de rôles à attribuer. Nous avons essayé de mettre un cadre dès le début et de nous attribuer des rôles précis (Denis était comme un "chef de projet" il s'occupait des commits et de la communication au sein de groupe et Hayaat agissait comme une développeuse) même si le but n'était pas de hiérarchiser notre groupe, bien au contraire, le but était d'essayer de respecter notre planning en prenant en compte toutes les bonnes pratiques de la conduite d'un projet pour se rapprocher au mieux du monde professionnelle et savoir prendre en compte le facteur du distanciel.

Je pense que nous avons su gérer le projet et prendre les bonnes décisions pour mener le objectifs à bien, le fait que ça soit un TER que nous avons proposé a été un moteur supplémentaire pour notre motivation et le fait de travailler souvent en binôme nous a permis de s'écouter et de faire l'effort de comprendre chacune des tâches demandés même si c'était le premier TER en distanciel cela reste une expérience enrichissante qui nous a permis de mettre en pratique ce que nous avons appris.

# Annexes



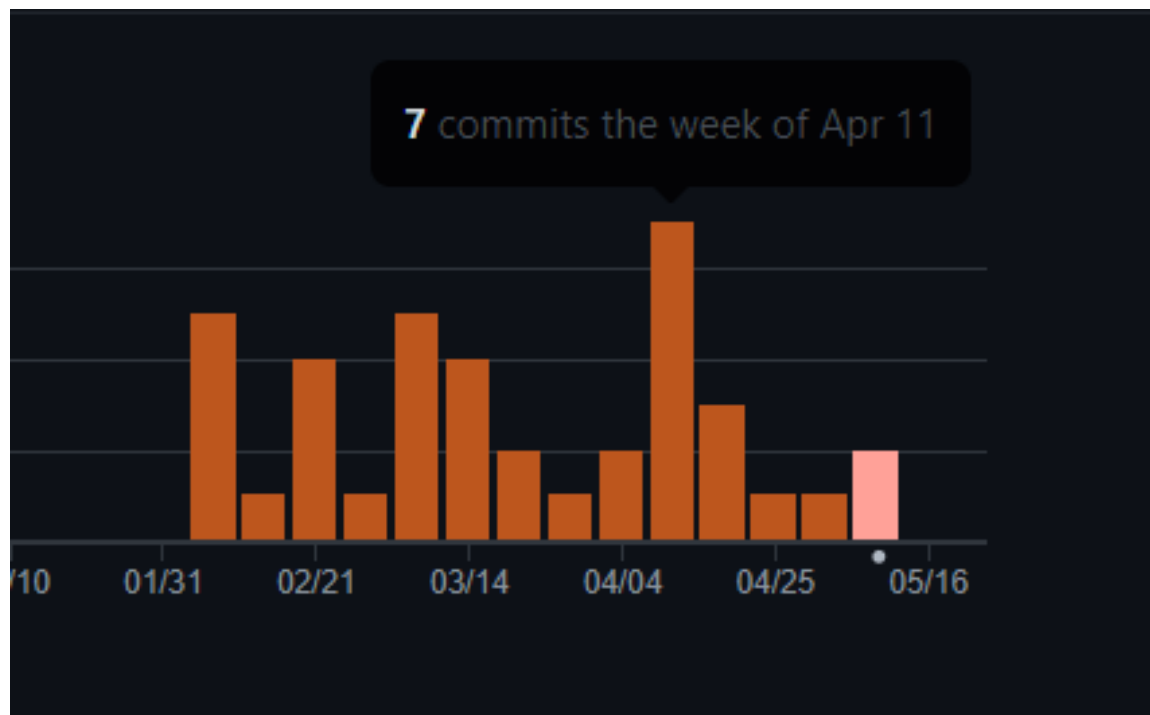


FIGURE 1 – Nombre de commit par mois

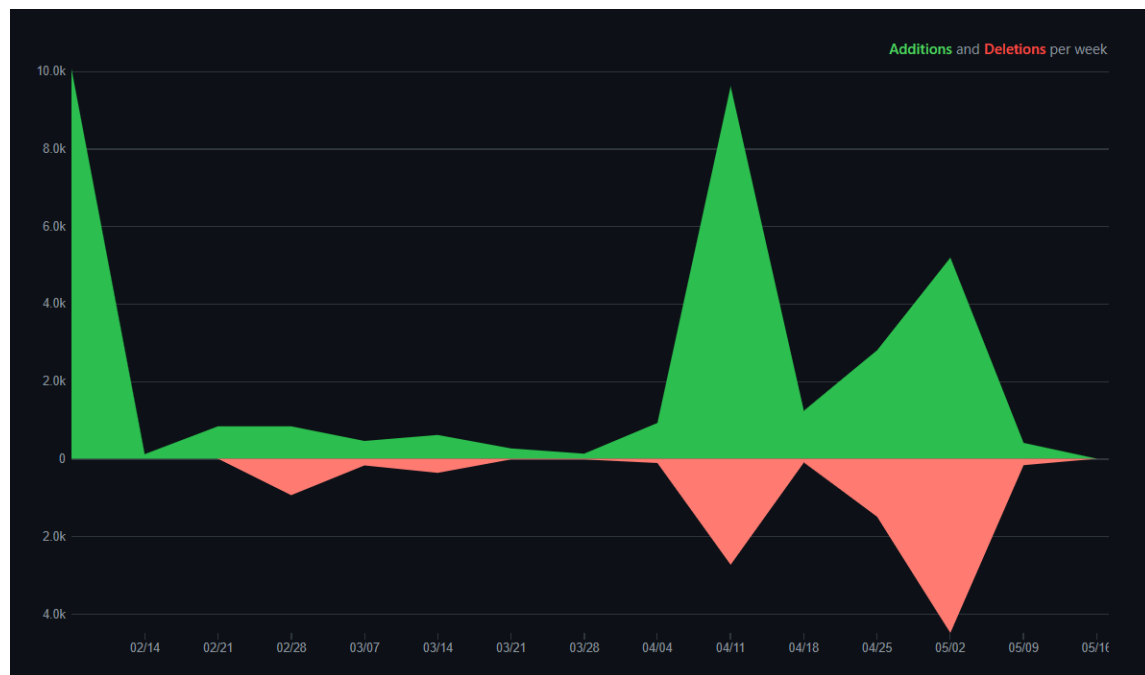


FIGURE 2 – Nombre d'additions et de deletions

**Intégrer des données provenant de différentes sources en présence d'une ontologie (lire : un modèle conceptuel des données en UML) afin de préparer les données à l'apprentissage et la prédiction.**

## 1 Les étapes

### 1.1 Chargement des données

La première étape consiste à charger les données des fichiers dans des bases de données adaptées. Nous optons pour deux systèmes, si possible différents, de base de données SQL pour les fichiers CSV ainsi qu'une base de données NoSQL pour le fichier JSON.

### 1.2 Définir le modèle ontologique

La seconde étape consiste à se mettre à la place des experts du domaine afin de définir un modèle ontologique (modèle conceptuel UML) permettant de décrire les données de façon unique.

### 1.3 Intégrer les données

La troisième étape consiste à intégrer les données des différentes sources dans le modèle ontologique (modèle conceptuel UML) en mettant en place un système de mappings de la forme

Requête sur la source  $-i$ , Données compatibles avec le format ontologique.

### 1.4 Graal

La quatrième étape consiste à utiliser Graal afin d'appliquer les mappings et les règles afin d'obtenir des données uniformes et complètes.

### 1.5 Apprentissage Prédiction / Visualisation

La cinquième étape consiste à déterminer qui va survivre au naufrage.

FIGURE 3 – Cahier des charges initial