



Ecole Polytechnique de l'Université de Tours
Département Informatique
64 avenue Jean Portalis
37200 Tours, France
Tél. +33 (0)2 47 36 14 14
polytech.univ-tours.fr

Projet Recherche & Développement 2021-2022

Entrepôt des données XML et visualisations de concepts sémantiques



POLYTECH[®]
TOURS

Entreprise

Polytech



Tuteur entreprise

Giorgia GALLUCI

Étudiant

Clément BOUTONNET (DI5)

Tuteurs académiques

Mohamed SLIMANE

Elena PIERAZZO

10 avril 2022

Liste des intervenants

Entreprise

Polytech
64 avenue Jean Portalis
37200 Tours, France
polytech.univ-tours.fr



Nom	Email	Qualité
Clément BOUTONNET	clement.boutonnet@univ-tours.fr	Étudiant DI5
Mohamed SLIMANE	mohamed.slimane@univ-tours.fr	Tuteur académique, Département Informatique
Elena PIERAZZO	elena.pierazzo@univ-tours.fr	Tuteur académique, Département Informatique
Giorgia GALLUCI	giorgia.gallucci@phd.unipd.it	Tuteur entreprise



Avertissement

Ce document a été rédigé par Clément Boutonnet susnommé l'auteur.

L'entreprise Polytech est représentée par Giorgia Galluci susnommé le tuteur entreprise.

L'Ecole Polytechnique de l'Université de Tours est représentée par Mohamed Slimane et Elena Pierazzo susnommés les tuteurs académiques.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assume l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable des tuteurs académiques et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



Pour citer ce document

Clément Boutonnet, *Entrepôt des données XML et visualisations de concepts sémantiques*,
Projet Recherche & Développement, Ecole Polytechnique de l'Université de Tours, Tours,
France, 2021-2022.

```
@mastersthesis{
  author={Boutonnet, Clément},
  title={Entrepôt des données XML et visualisations de concepts sémantiques},
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université de Tours},
  address={Tours, France},
  year={2021-2022}
}
```

Table des matières

Liste des intervenants	a
Avertissement	b
Pour citer ce document	c
Table des matières	i
Table des figures	iv
1 Introduction	1
1 Acteurs, enjeux et contexte	1
2 Objectifs	1
3 Hypothèses	1
4 Bases méthodologiques.....	2
2 Description générale	3
1 Environnement du projet	3
2 Caractéristiques des utilisateurs.....	3
3 Fonctionnalités du système	3
4 Structure générale du système.....	3
3 État de l'art / Veille technologique	5
1 L'édition numérique	5
Exemple d'éditions numérique	5
Outils pour une éditions numérique	5
La visualisation	6

4	Analyse et conception	7
1	Analyse	7
1.1	Spécifications.....	7
2	Modélisation proposée.....	7
5	Bilan et conclusion	8
1	Bilan du semestre 9	8
2	Bilan du semestre 10.....	8
3	Bilan sur la qualité	12
4	Bilan auto-critique.....	12
	Annexes	13
A	Planification, gestion de projet	14
1	Evolution du projet	14
2	Description des tâches.....	14
	Tâche 1 : Intitulé parlant	14
	Tâche 2 : Intitulé parlant	15
B	Description des interfaces	16
1	Interfaces matérielles/logicielles	16
2	Interfaces homme/machine.....	16
C	Cahier de Spécifications	17
1	spécifications Fonctionnelles.....	17
1.1	Fonctionnalités à développer	17
2	Spécifications non fonctionnelles	18
2.1	Contraintes de développement et conception	18
2.2	Contraintes de fonctionnement et d'exploitation.....	18
2.2.1	Performances.....	18
2.2.2	Capacités	18
2.2.3	Contrôlabilité	18
2.2.4	Sécurité	18
D	Cahier du développeur	19
1	Introduction	19
2	Descriptions détaillées de données exploitées	19
E	Document d'installation	20
F	Document d'utilisation	21

G	Cahier de test	22
1	Tests unitaires	22



Table des figures

2	Description générale	
2.1	Diagramme des cas d'utilisation.....	4
3	État de l'art / Veille technologique	
3.1	Fouille de données et visualisation.....	6
3.2	Fouille de données et visualisation.....	6
5	Bilan et conclusion	
5.1	Interface odd TEIPublisher.....	9
5.2	Rendu de l'odd.....	9
5.3	Exemple de base avec la légende.....	11
5.4	Exemple de choix des paramètre avec le menu déroulant.....	11
5.5	Exemple de l'affichage des occurrences.....	11
A	Planification, gestion de projet	
A.1	Le diagramme de Gantt Final.....	14

1

Introduction

1 Acteurs, enjeux et contexte

Présentation du contexte des acteurs et des enjeux.

Acteurs :

- Client : Giorgia Galluci
- MOA : Elena Pierazzo et Mohamed Slimane
- MOE : Clément Boutonnet et Jules Nuguet

2 Objectifs

Nous avons deux objectifs à réaliser pour ce projet. Premièrement créer un entrepôt pour la gestion du corpus des données, nous allons baser le nôtre sur une DB-XML avec eXistDB qui prend des données sous format xml. Nous allons utiliser TeiPublisher comme espace de travail car c'est ce qui a été conseillé par le client et cet espace de travail à vocation à être utilisé pour des projets de science humaine. Le deuxième objectif est d'élaborer une interface utilisateur ergonomique pour la visualisation et la recherche des données pour des non spécialistes. Pour résumer, nous devons réaliser une édition numérique innovante en sciences humaines. Il faudra donc faire en sorte d'avoir une interface utilisateur (visualisation) la plus attrayante et innovante possible tout en pensant le projet pour que ce dernier puisse être réutilisé pour d'autres corpus. Les textes de ce corpus sont fournis par Mme Galluci.

3 Hypothèses

Les différentes hypothèses de ce projet sont :

- Membres d'équipe malades / problèmes personnels : rééquilibrer la charge de travail, voir déplacer les partis moins critiques à plus tard et proposer une aide personnelle.
- Spécifications incomplètes : demander plus de précisions.
- Problèmes de fiabilité et robustesse : Travailler davantage sur la conception vers la fiabilité et la modularité. Effectuer sérieusement les tests et générer la doc dont les différents diagrammes pour mieux analyser l'architecture.

- Démotivation de l'équipe : prévoir des temps de pause, surtout en période d'exams et prendre en compte les autres projets à côté.
- Mauvaise estimation du temps / coût : Effectuer des mises à jour des estimations fréquemment. Prévoir une petite marge d'erreur sur la date finale.
- Plusieurs sauvegardes sur des plateformes différentes dans le cas d'incidents / accidents. Si l'installation du serveur se passe mal, dans le GANTT cette action est placée en amont du projet pour pouvoir gérer cette probabilité.
- Problème informatique qui empêche la communication avec Mme Galluci (qui sera en Italie au second semestre) => réunion fréquente avec l'équipe et présentation des résultats pour ne pas avoir de retard à cause de ce problème.

4 Bases méthodologiques

Pour ce projet nous avons choisi une méthodologie agile car nous avons la chance d'être en relation direct avec Mme Galluci et cette méthode nous permettra d'avoir son avis à chaque étape du projet, ce dernier étant important pour elle car c'est son sujet de thèse. Ainsi elle pourra nous faire remonter chacun point à améliorer ou nous exprimer plus en détail sa vision finale du projet avec un exemple à chaque fois. Comme elle va nous fournir les différents livrets d'opéras composant le corpus au format XML, il est important de lui montrer comment vont rendre les différents textes via TeiPublisher.

La fonction de recherche étant une fonction très importante pour elle, elle pourra nous faire part de ses envies à mesure que le projet avance.

Nous avons aussi des réunions chaque semaine avec toute la team pour voir les avancées du projet et de se rendre compte si ce dernier part dans le mauvais sens.

Nous allons donc nous occuper de réaliser dans un premier temps des fonctions de recherches basiques puis nous allons les améliorer au fil du temps. De plus pour la partie visualisation nous allons réaliser la frise petit à petit en ajoutant les différentes options au fur et à mesure. De même pour la présentation attractive des textes.

Nous allons mettre en place un Trello avec Mr Nuguet pour ne pas se perdre et rester dans les temps conformément au Gantt. Nous allons aussi réaliser des tests à chaque fin de sprint pour vérifier que tout fonctionne correctement.

Nous avons mis en place un Google Drive que nous mettons à jour régulièrement pour pouvoir suivre les différentes mises à jour du projet.

Nous allons rendre un document expliquant le fonctionnement des différentes fonctionnalités, comment les améliorer et comment les utiliser simplement.

2

Description générale

1 Environnement du projet

Le projet final sera intégré sur un serveur de l'université de Padoue, ce dernier est géré par Marco CASADEI avec qui nous avons pu échanger via Teams.

2 Caractéristiques des utilisateurs

Les utilisateurs seront des personnes qui ne s'y connaissent pas trop en informatique, nous devons donc faire des interactions simples et intuitives qui leur permettront de réaliser toutes les tâches qu'ils souhaitent. De plus, il est prévu de rendre un livret d'utilisation complet et très détaillé.

L'interface sera de même très intuitive comme la fonction de recherche pour permettre à tous de pouvoir rechercher toutes les informations qu'ils souhaitent sur les opéras composant le corpus. Il y aura des droits à créer car il y aura des administrateurs qui pourront déposer des textes et des personnes validant ces derniers.

3 Fonctionnalités du système

Décrire les fonctions utilisateurs du système et les principaux objets qui constituent le système. Ceci peut être fait au travers d'un ou plusieurs diagrammes de cas d'utilisation généraux par type d'utilisateur.

4 Structure générale du système

Identifier les principaux composants/éléments du système ainsi que leurs relations. Cela peut être fait au moyen de diagrammes statiques (« objet » au sens large, i.e. classe, composant, déploiement, etc.). Sans être une analyse à part entière, cette partie doit montrer que vous avez une première réflexion sur la structure interne.

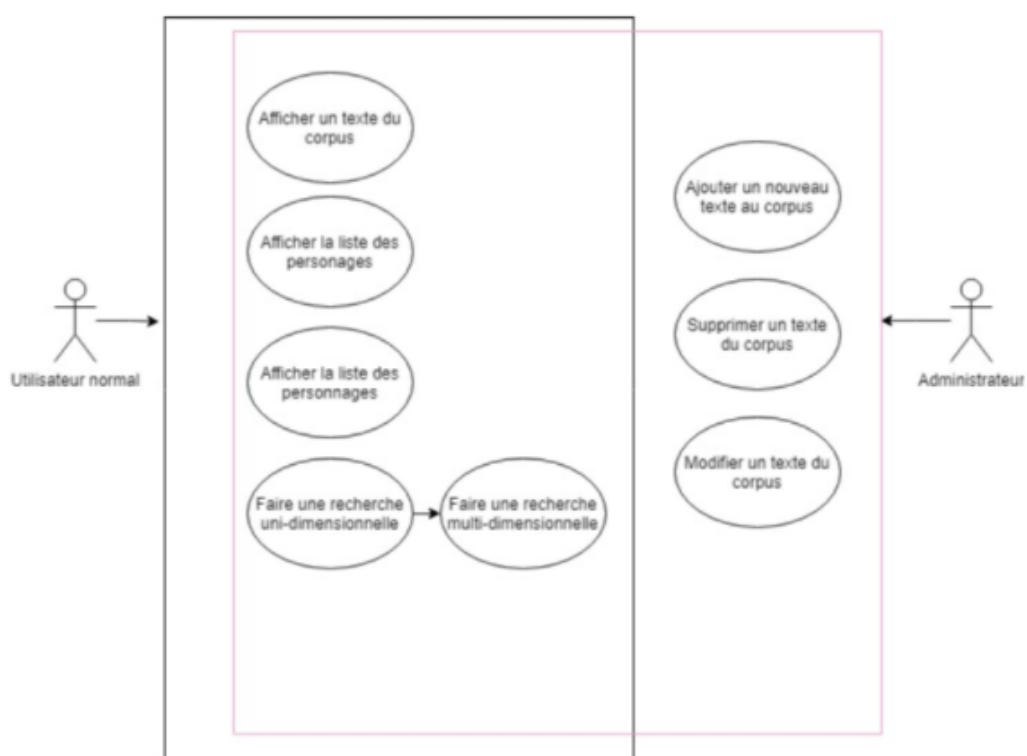


Figure 2.1 – *Diagramme des cas d'utilisation*

3

État de l'art / Veille technologique

1 L'édition numérique

Exemple d'éditions numérique

Une édition numérique dans le milieu universitaire a pour objectif de rendre publique un corpus de source primaire à un public large. Une part importante du travail consiste à appliquer une série de balisage d'un texte brut pour le rendre intelligible pour l'humain. Dans le domaine des Humanités le standard massivement employé est le standard TEI, mais il en existe d'autres moins répandus. La première partie dans la création d'une édition numérique critique est donc de créer un processus de balisage pour mettre en évidence ce qu'on souhaite analyser. Une grande partie réside donc aux choix éditoriaux du balisage en fonction de l'objectif du projet.

Il existe aujourd'hui de nombreux projets qui diffusent publiquement leur corpus avec des niveaux d'analyse très hétéroclites. Quand certains projets comme l'édition numérique des Misérables proposent de mettre en évidence l'évolution d'un texte entre les différentes versions d'un manuscrit. D'autres proposent des entrées avec des champs de recherche permettant de mettre en valeur une unité du corpus comme c'est le cas dans le projet RENOM. Souvent les éditions numériques proposent des champs de recherche pour mettre en évidence les métadonnées du texte ou les entités nommées. On pourrait également citer L'édition des textes de Charles de la Fontaine ou l'édition des textes de Marc Michel Rey. Ces deux projets ont une ergonomie assez semblable avec des index séparés du contenu du texte. Dans une édition numérique la part de l'interface joue grandement dans l'évaluation qu'on peut en avoir. On peut notamment penser à l'édition Savoirs, où l'on remarque une intention de réflexions sur les interactions Hommes-machines pour rendre la recherche didactique. Le contexte d'un projet joue une importance dans la mise en valeur d'un élément ou un autre correspondant à la source primaire et à l'analyse qu'on souhaite y porter. Une réflexion importante à notamment élever est de mener sur le projet BONHUM

Outils pour une édition numérique

D'un point pratique il existe de nombreux outils pour mettre en place une édition numérique sur le choix de l'outil pour la base de données. Au vu des données que nous avons on préférerait une base de données NoSql comme BaseX, MongoDB ou moins connus eXist.

Le choix d'eXist semble le plus pertinent car il existe déjà de nombreux outils reliés à la TEI grâce à l'application TEI Publisher. Il existe à ce jour quatre projets d'édition numérique avec

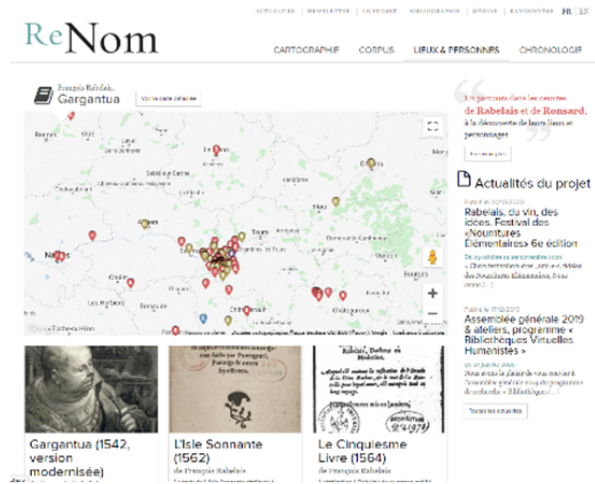


Figure 3.1 – *Fouille de données et visualisation*

ces outils (Shakespeare , les lettres de Van Goth Letters , Document sur la chute du mur de Berlin et des livres Anglais). Il possède tous des champs de recherches et la consultation de ressource mais aucune visualisation graphique du corpus.

La visualisation

Aujourd'hui dans le domaine des humanité numérique un courant de pensée visant à mélanger une édition numérique assez standard à des outils de distant reading se développe. Le distant Reading consiste à analyser un texte en prenant de la hauteur pour étude analytique. Ce qui diffère entre une édition numérique et une étude de distant reading c'est éditorialisation du contenu. Comprendre les concept distant reading avec de l'éditorialisation permet la création d'algorithmique d'analyse, de suggestion et de visualisation plus facile dans la lecture d'un corpus. Certains éditions Numérique on tenter. Il existe quelques éditions numériques qui possèdent des visualisations comme Perseus Digital.

Il existe de nombreux outils permettant de nous inspirer sur la visualisations : voyant tools , datavizproject et histography.io .

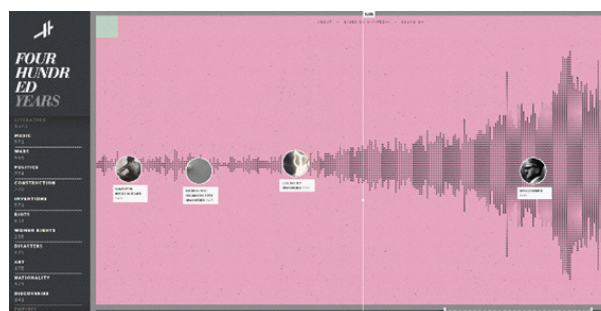


Figure 3.2 – *Fouille de données et visualisation*

4

Analyse et conception

1 Analyse

1.1 Spécifications

Voir le cahier des spécifications

2 Modélisation proposée

Inclure ici une description du système à développer pouvant notamment inclure les principaux diagrammes UML non détaillés et démontrant sa faisabilité durant la phase de mise en œuvre.

Les modes de validation prévus pour les différents éléments à produire pourront être précisés ici.

5

Bilan et conclusion

1 Bilan du semestre 9

Tout s'est bien déroulé pendant le semestre 9. En effet, j'ai pu faire de nombreuses réunions avec Mme Galluci qui m'a permis d'avoir une bonne connaissance du sujet pour pouvoir avancer rapidement et efficacement. Nous avons aussi été qualifié au concours de TEI Publisher dans le but de faire partie de leur page de présentation. Ils ont été intéressés par notre projet et nous aurons une réunion le 14 janvier pour pouvoir parler d'avantage avec eux. Nous avons la chance d'avoir été qualifié au côté de 3 autres équipes. Nous allons aussi devoir préparer une présentation de notre projet pour la DH2022 qui se déroulera au Japon. C'est aussi une bonne nouvelle car ce rassemblement d'humaniste peut être une bonne chance de créer des contacts pour plus tard. C'est donc sur ces deux bonnes nouvelles que nous avons appris à la fin du semestre que nous finissons celui-ci.

2 Bilan du semestre 10

Ce projet était spécial il était en collaboration avec le CESR. J'étais donc avec un autre étudiant Jules Nuguet, nos superviseurs académique était Mr Slimane et Mme Pierazzo et notre client était Mme Galluci.

TEIPublisher :

Nous devons pour cette partie créer un odd qui allait présenter les différents textes du corpus et pour se faire Mme Pierazzo nous a conseillé d'utiliser TEIPublisher. J'ai rencontré plusieurs problèmes sur cette partie consistant à une visualisation des textes xml. Tout d'abord je ne peux pas installer docker sur mon pc car il était ancien. J'ai d'abord installé une version existDB pour avoir accès à ma propre version de TEIPublisher mais là encore je ne pouvais pas installer compatible avec TEIPublisher, je suis donc allé sur leur site en ligne et je me suis connecté via la version demo de leur site pour pouvoir réaliser cet odd. Premièrement, il a fallu comprendre comment fonctionne un odd mais le site de TEIPublisher est très bien fait. En effet, j'ai recherché sur internet comment réaliser cet odd mais j'ai aussi regardé les odds des autres personnes utilisant la version demo pour m'inspirer ou apprendre plus sur tous les changements possibles. Le site permet de visualiser les changements qu'on fait de façon simple et rapide ce qui est bien pour nous. Malheureusement nous avons eu un problème car dans un premier temps, l'importation des textes de Mme Galluci sous TEIPublisher ne fonctionnait pas. En effet, il est dur de travailler

sur la visualisation des textes quand on ne peut pas voir ces derniers. Mais après quelques temps, Jules a réussi à débloquent la situation et j'ai de mon côté profiter de ce temps pour en apprendre plus sur comment coder un odd. Voici comment est représenté l'interface de création d'un odd dans TEIPublisher, c'est facile d'utilisation quand on sait ce que l'on fait et qu'on connaît bien les balises xml à modifier :

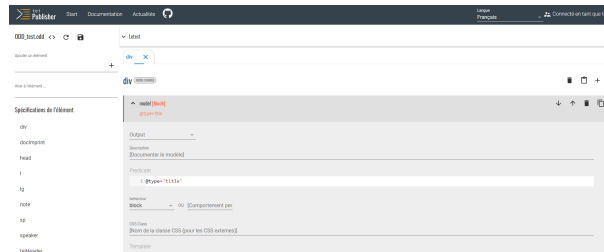


Figure 5.1 – *Interface odd TEIPublisher*

Au final voici une partie d'un texte de Mme Galluci avec l'odd final :

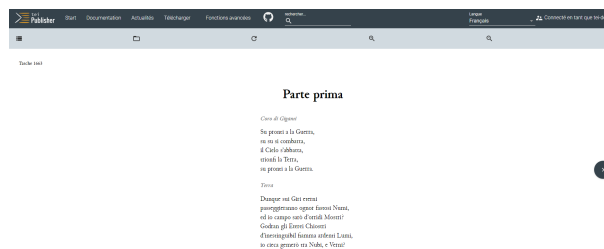


Figure 5.2 – *Rendu de l'odd*

Pour valider cet odd nous avons eu plusieurs réunions avec Mme Galluci durant lesquelles nous lui avons montré les avancements jusqu'à ce qu'elle valide la version finale. Nous lui avons aussi expliqué via un exposé en distanciel comment facilement transformer l'odd si jamais elle changeait d'avis sur certaines visualisations.

Frise chronologique :

Le doctorat de Mme Galluci porte sur l'évolution de différentes idées au cours de plusieurs textes d'opéras. Notre travail était de trouver une idée innovante et simple d'utilisation pour que chacun puisse voir l'évolution de ces idées au cours du temps. Nous sommes donc partis sur l'idée d'une frise chronologique qui allait présenter les éléments de façon interactive. Les textes de Mme Galluci sont au format xml et nous avons de choisir de partir sur une page html ainsi qu'un javascript pour gérer la création des points représentant les différents items que l'on doit gérer. La première chose que j'ai faite était de regarder différents exemples de frise chronologique sur internet pour pouvoir m'en inspirer et créer une frise qui nous correspondait. Les frises trouvées ne présentaient pas forcément ce que je voulais car toutes les dates étaient créées dans le fichier html et je préférais les avoir dans le fichier javascript car cela permettait une reprise du code plus simple et une facilité pour changer les différentes informations. Je suis donc parti sur une boucle pour la création de la frise à partir d'une date de début et de fin donnée en variable globale pour faciliter les changements. Ces dates sont aussi utilisées dans d'autre partie du code. J'ai aussi ajouté un pas en variable globale pour choisir l'écart entre deux dates (si on veut une frise chronologique qui va de 5ans en 5 ans par exemple). Tout cela me créer une liste de date qui est géré et affiché par le fichier html. Je me suis ensuite occupé d'afficher la frise au milieu de la page pour permettre d'avoir 2 zones distinctes (une en haut et une en bas) permettant d'afficher des informations séparées les unes des autres. Ensuite il a fallu s'occuper de l'apparition et du positionnement des points. Pour ce faire j'ai déjà créé deux patrons, un pour en haut et un pour en bas. Ainsi, j'ai créé soit des points soit des carrés qui seront ensuite utilisé pour représenter les différents items. Pour permettre en reprise du code facile j'ai mis c'est deux forme

en tant que variable globale sous le nom, formehaut et formebas. Ainsi si on veut changer une forme, il suffit juste d'ajouter son patron dans le fichier css puis de mettre le nom de la forme comme valeur à formehaut ou formebas pour que cela change directement la visualisation. Il faut aussi placer les points au bon endroit. Pour se faire il suffit de retrouver la position de la première date puis de faire en sorte d'ajouter à chaque fois l'écart de la date actuel à celui de la première date pour placer parfaitement les points. On fait ce même principe pour le placement sur l'axe vertical et ainsi les points sont bien placés. J'avais pour l'instant juste fait des tests avec une création aléatoire des points pour tester mon programme. Ensuite il fallait connecter les fichiers xml au javascript pour créer les points en fonction des textes. J'ai beaucoup cherché pour cette partie car j'ai essayé plusieurs méthodes qui ne fonctionnaient pas dans un premier temps mais j'ai réussi à trouver grâce au XMLContent et au fetch. Via un parser on va transformer un fichier xml donné en groupe de node puis grâce à l'installation de node.js on peut retrouver les informations contenues dans des nodes grâce à querySelector. Je m'occupe dans un premier temps de récupérer la date du texte pour vérifier que celui-ci est bien compris entre la date de début et de fin défini et donc peut apparaître sur la frise. On récupère la date sous forme de string mais avec un parser on la transforme en int pour ne pas avoir de soucis. Pour récupérer les items important par la suite on doit regarder l'attribut que leur à donner Mme Galluci car cela permet de les traiter plus facilement. Les informations à récupérer se trouvent toujours dans les mêmes balises. J'ai quand même mis le nom de ces champs en variables global pour qu'ils soient plus faciles à changer. On récupère ainsi le nom des personnages, des références aux auteurs de l'antiquité, le nom des speakers et les concepts. Ensuite toutes ces valeurs sont mises dans une liste. Pour chaque année on enlève les doublons puis on remet dans une liste comme ça on obtient une liste finale où l'indice va représenter les années. Prenons l'année de début, je vais trouver tous les items présent dans le textes au niveau de la dernière liste si j'utilise l'indice 0 (ListeFinale[0]). Si j'utilise l'indice 1 j'aurai les items de l'année du début +1 et ainsi de suite. C'est bien plus facile de se retrouver comme ça et ça évite les erreurs. Ensuite on va créer les points, après être passé dans une boucle pour vérifier que tous les traitements sont bien terminés (le fetch est assez compliqué sur ce point ce qui m'a attiré pas mal d'erreur), on va créer un nouvelle élément qui sera le point à créer auquel on va donner toutes les informations importantes puis l'afficher sur le graphique comme vu précédemment. On va aussi donner sa couleur au point en fonction de la valeur qu'il représente. A ce stade il ne manquait plus qu'à ajouter une légende présentant à quoi correspondaient les points. Pour ce faire j'ai dans un premier temps écrit la légende dans une liste puis je l'ai placé directement dans le html. Mais je n'étais pas content car ce n'était pas propre. Puis j'ai repensé à la création d'élément et je suis passé par une création directement en javascript qui est beaucoup plus facile à changer. Un autre problème m'embêtait encore, j'avais fait une liste avec les textes xml de giorgia mais cette liste était en dur et donc je trouvais ça pas propre. J'ai fait beaucoup de recherche sur le sujet et j'ai vu que pour des raisons de sécurité on ne pouvait récupérer une liste de fichier via un code javascript. Mais après encore plus de recherche et beaucoup de test (vraiment beaucoup) j'ai trouvé sujet qui disait rusé et pouvoir faire en sorte de trouver tous les dossiers appartenant à une certaine extension (ici xml). J'ai essayé et après quelques modifications tout fonctionne. Je vais être honnête je n'ai pas compris tout ce bout de code et je suis incapable de le refaire entièrement moi-même mais j'ai réussi à le modifier pour qu'il fonctionne pour une extension xml. De ce que j'ai compris, ce code prend la page html la transforme en node et cela permet de retrouver les fichiers xml pourtant présent dans un autre dossier. Un problème subsistait encore à ce stade, je supprimais les doublons, si un item était présent plusieurs fois dans un texte, je ne le présentais qu'une seule fois. Après discussion avec Jules nous sommes venus à penser à 3 solutions : - La position du point se ferait en fonction du nombre d'apparition dans le texte ; - Le rayon du point serait en fonction de son nombre d'apparition ; - On affiche le nombre d'apparitions de l'item sur le point ; La première solution présente un problème si beaucoup de points ont le même nombre d'apparition car on perd de l'information. La deuxième solution est aussi problématique, elle fonctionne mal pour les carrés et certaines années pourrait se superposer. Je suis donc parti sur la dernière proposition

et j'ai associé à chaque point un système d' « addEventListener » qui prend en compte quand la souris passe sur un point ou en sort. J'ai aussi fait en sorte que passer sur un point situé en haut affiche le nombre d'itération de tous les points du haut et pareil pour en bas. J'ai fait cela en pensant à l'utilisateur qui voudra surement comparer les points de même couleur et donc il n'aura pas à regarder tous les points un par un. J'ai ajouté une liste d'occurrence qui, avant la suppression des doublons, va associer pour chaque valeur son nombre d'apparition. Nous avons 4 familles d'items à ce stage, les personnages, les références aux textes antiques, les concepts et les speakers. Ces 4 familles ne doivent pas être affichées en même temps sur la frise donc j'ai créé une liste déroulante où l'on peut choisir la famille que l'on veut afficher. J'ai dû changer pour cela le code et donner un indice pour chaque famille. J'ai aussi créé un indice pour le haut et le bas et avec ces ajouts plus quelques changements je peux afficher la famille désirée. J'ai toujours un petit problème car si l'on affiche deux fois la même famille il y a un problème sur les couleurs de la légende, ces dernières ne sont pas affichées, je sais à quoi cela ait dû mais je ne vois pas comment changer avec le peu de temps qu'il me reste, j'ai donc juste affiché une alerte disant de ne pas choisir deux champs identique à la place. Pour finir, dans chaque texte il y a 3 sous-parties comme on pourrait dire, et Mme Galluci souhaitait que cela apparaisse sur la frise. J'ai donc changé mon code une dernière fois pour faire en sorte ces trois parties soit bien prises en compte dans le code. Cela a été compliqué même en comprenant ce qu'il fallait faire mais j'ai finalement réussi après beaucoup d'essais la fonction de « node.contains » qui permet de voir si une node en contient une autre. Le résultat final de la frise est le suivant, j'ai passé énormément de temps dessus et je suis satisfait au final :

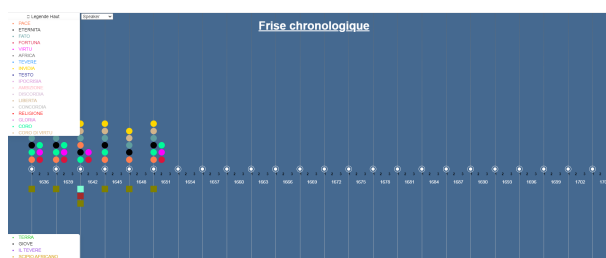


Figure 5.3 – Exemple de base avec la légende

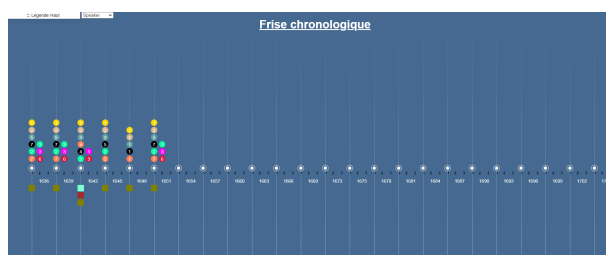


Figure 5.4 – Exemple de choix des paramètres avec le menu déroulant

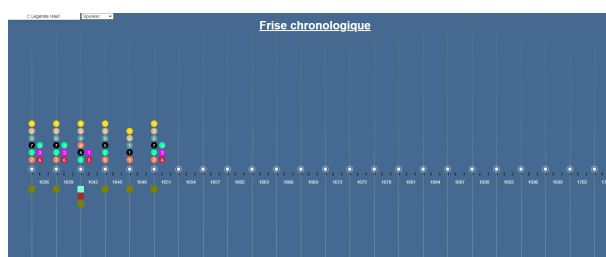


Figure 5.5 – Exemple de l'affichage des occurrences

Intégration site et serveur italien :

Je n'ai pas eu le temps de finir cette partie car la frise m'a pris plus de temps de prévu pour faire chaque fonctionnalité de cette dernière. Je ne pense pas que ce soit trop difficile, j'ai commencé à avancer dessus mais le manque de temps couplé à mon manque de connaissance sur ce sujet font que je ne peux pas terminer cette partie.

3 Bilan sur la qualité

J'ai essayé de faire un effort maximum sur la qualité du code en mettant le plus possible des variables globales pour permettre une maintenance et un changement facile si besoin. J'ai commenté le code de façon claire et précise. Le nom de chaque variable est bien défini à part le nom des int des boucles for où je suis resté sur les traditionnelles lettres (i,j...). Chaque fonction et chaque partie du code est expliquée plus en détail dans le guide utilisateur disponible sur le git. Je pense avoir rendu un code assez clair et maintenable et qui peut supporter des changements si une autre personne veut s'inspirer de ce projet. Je suis au final globalement satisfait de la qualité du code.

4 Bilan auto-critique

Je pense que mes spécifications n'ont pas été précises et cela a entraîné un petit retard qui est de ma responsabilité. En effet, j'ai dû faire beaucoup de changements la dernière semaine car Mme Galluci nous a demandé de le faire. J'aurais dû me faire confiance et m'écouter sur certains points pour éviter ce retard. Je pense avoir beaucoup travaillé sur ce projet et m'être beaucoup investi sur ce dernier pour rendre un résultat aussi proche que possible de ce qu'avait en tête la cliente. Je suis un peu déçu de ne pas avoir complètement fini mais je suis content du rendu car je pense avoir fait le maximum avec mes compétences. J'ai beaucoup appris pendant ce projet car j'ai fait énormément de tests, j'ai fait face à beaucoup d'erreurs et j'ai fait énormément de recherches sur les différents points. Je n'avais jamais fait de javascript et le seul cours d'html que j'avais eu était en PEIP donc je pense que le résultat est bon. Je pense devoir faire encore des efforts pour chiffrer correctement le temps qu'une tâche me prend car je n'ai pas été assez précis.

Je ne suis normalement pas très travailleur mais je pense que j'ai prouvé avec ce projet que je pouvais beaucoup travailler même sur mon temps libre pour mener ce dernier. Je remercie Mme Pierazzo et Mr Slimane qui m'ont obligé à un travail quotidien que ça soit via les réunions ou les rapports de semaine.

Annexes

A

Planification, gestion de projet

1 Evolution du projet

Le diagramme de Gantt Initial pour la planification de ce projet

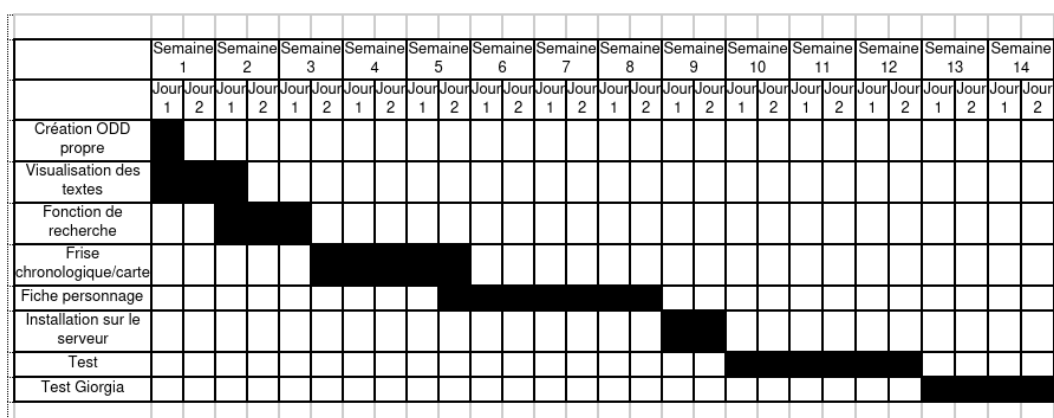


Figure A.1 – Le diagramme de Gantt Final

2 Description des tâches

Tâche 1 : Intitulé parlant

- Date de début : 16/09/2019
- Date de fin : 03/10/2019
- Durée : 17 jours
- Description : éléments à faire, liste des entrées (pré-requis) et sorties (livrables)s de la tâche.

Tâche 2 : Intitulé parlant

- Date de début : 16/09/2019
- Date de fin : 03/10/2019
- Durée : 17 jours
- Description : éléments à faire, liste des entrées (pré-requis) et sorties (livrables)s de la tâche.

B

Description des interfaces

TeiPublisher est un outil créé pour les humanistes leur permettant de publier leurs travaux sans pour autant qu'ils aient des connaissances poussées en informatique et en programmation. TeiPublisher supporte des documents au format XML et nous pouvons utiliser XQuery comme langage de requête.

1 Interfaces matérielles/logicielles

Nous utilisons TeiPublisher pour gérer la base de données, ainsi tout se déroulera via Internet et le site de l'université de Padoue.

2 Interfaces homme/machine

Tout le monde peut se connecter au site de l'Université de Padoue pour voir le travail que nous aurons effectué. Pour se déplacer sur le site il suffira d'utiliser une souris et d'utiliser les différents champs de recherche disponible pour parcourir tous les opéras répertoriés plus facilement.

C

Cahier de Spécifications

1 spécifications Fonctionnelles

Il s'agit de l'expression des besoins fonctionnels. Cette partie a donc comme objectif de décrire l'ensemble des fonctions du système en précisant avec quels composants elles interagissent. Des diagrammes de cas d'utilisation plus détaillés, ainsi que l'arbre hiérarchique des fonctionnalités pourra être fourni ici pour donner une vision plus globale. En outre, chaque fonction sera décrite précisément (cf. ci-dessous). Là encore, il s'agit d'une pré-analyse indispensable à l'évaluation de la complexité de votre projet et à la planification de sa réalisation.

1.1 Fonctionnalités à développer

Recherche simple : Affiche les résultats par nombre croissant d'entité dans chaque texte de l'item recherché (personnage, idée...). Reprendre et améliorer la fonction de recherche déjà existante dans TEIPublisher.

Recherche multiple : Après avoir réalisé une recherche simple, il est possible de continuer de faire des recherches simples considérant seulement les textes obtenus. Affiche les résultats par nombre croissant d'entités combinées dans chaque texte. Reprendre et améliorer la fonction de recherche déjà existante dans TEIPublisher.

Création d'une frise chronologique : Frise chronologique où les différents textes seront placés avec possibilité de cliquer sur la frise pour arriver sur le texte en question.

Création carte géographique : Carte où les différents lieux sont indiqués par des points et où on peut recevoir les informations des différents en cliquant dessus.

Visualisation des textes : Reprendre et améliorer les exemples existants sur TEIPublisher comme les lettres de Van Gogh.

Fiche personnage : Réaliser une fiche pour chaque personnage résumant les idées qu'ils incarnent en fonction des textes, les textes où ils sont présents et leurs caractéristiques si ces dernières sont semblables pour tous les opéras.

ODD : Nous allons créer un ODD propre au projet à partir d'un déjà existant. Un ODD permet de gérer la visualisation sur le site.

2 Spécifications non fonctionnelles

2.1 Contraintes de développement et conception

Matériel : un ordinateur avec une connexion internet

Langage : Une base de données de type XML avec les textes en xQuery.

Environnements nécessaires : TeiPublisher

2.2 Contraintes de fonctionnement et d'exploitation

2.2.1 Performances

Afin de ne pas gêner l'expérience de l'utilisateur, nous voulons des temps de réponse courts en termes de recherche ou de multi-recherche. Normalement la vitesse des recherches via TeiPublisher sont satisfaisantes mais voir s'il ne vaut pas mieux gérer ces dernières en fonctions du nombre de texte que nous allons ajouter. du point de vue de l'utilisateur : temps de réponse souhaité, fréquence d'utilisation, temps d'indisponibilité acceptable, etc. ;

2.2.2 Capacités

Même chose ici que le point précédent, la quantité de stockage proposé par TeiPublisher est suffisante pour ce projet.

2.2.3 Contrôlabilité

Nous devons contrôler les différents comptes, administrateurs, les comptes valideurs et les comptes simples utilisateurs.

2.2.4 Sécurité

Les textes du corpus sont disponibles en ligne sur internet donc il n'y pas de sécurité nécessaire à ce niveau mais il faudra sécuriser les documents XML si Mme Galluci le veut pour pas que certaines personnes puissent voler son travail (à revoir avec elle). Vérifier que les livrets ajoutés ne contiennent pas de données pouvant « détruire » le site. Les données personnelles des utilisateurs sont normalement gérées par TeiPublisher.

D

Cahier du développeur

1 Introduction

Le projet devait comporter 3 grandes parties : - Une partie sur la visualisation des textes via TEIPublisher et la création d'un odd. Un odd est un fichier de visualisation de texte spécifique à TEIPublisher. - La deuxième partie se portait sur la production d'une frise chronologique présentant divers éléments et items que l'on retrouvait dans les textes du corpus. - La dernière partie qui était l'intégration du projet sur le serveur italien. Je n'ai malheureusement pas pu terminer la troisième partie par manque de temps et de connaissance sur le sujet.

Voir la partie bilan pour comprendre le déroulé du projet

2 Descriptions détaillées de données exploitées

Les données exploitées sont des textes au formats xml fournis par la cliente Mme Galluci.

E

Document d'installation

Si on ne prend pas en compte l'intégration sur le serveur italien qui est en cours :

-On doit utiliser Visual Studio Code -Installer l'extension Live Server –Avoir node.js installé

Il suffit de cliquer droit => "Open with Live server" depuis la page html.

A blue square containing a white capital letter 'F'.

Document d'utilisation

A voir sur le git



Cahier de test

Les tests visent à garantir l'exactitude, l'intégrité, la sécurité et les performances du logiciel.

1 Tests unitaires

Toutes les fonctions ont été testé plusieurs fois. Notamment le fait d'avoir plusieurs texte pour une même année, cela affiche bien deux fois les points avec les bonnes occurences.

Entrepôt des données XML et visualisations de concepts sémantiques

Clément Boutonnet

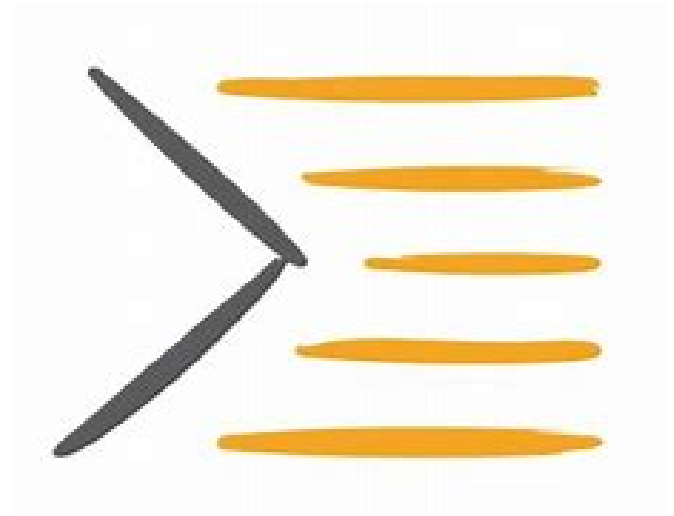
Encadrement : Mohamed Slimane et Elena Pierazzo



En collaboration avec Polytech

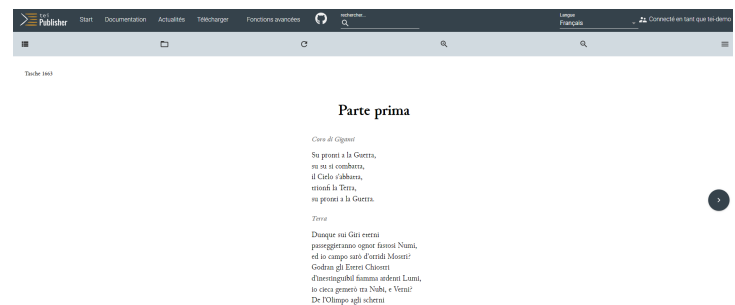
Objectifs

- Visualisation des textes : TEIPublisher
- Frise chronologique
- Intégration sur le serveur italien



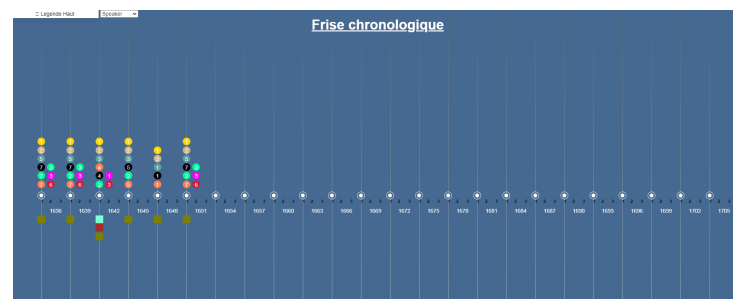
Mise en œuvre

1. Création d'un odd
2. Création d'un html et d'un javascript
3. Actuellement sur serveur local



Résultats attendus

Voici la frise chronologique finale :



Entrepôt des données XML et visualisations de concepts sémantiques

Clément Boutonnet

Encadrement : Mohamed Slimane et Elena Pierazzo



En collaboration avec Polytech

Objectifs

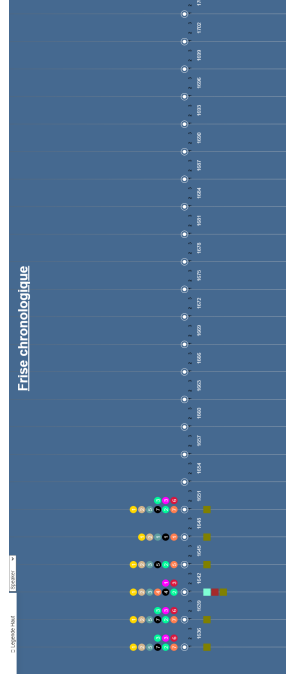
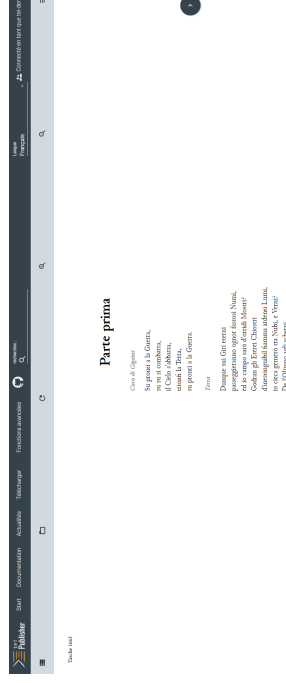
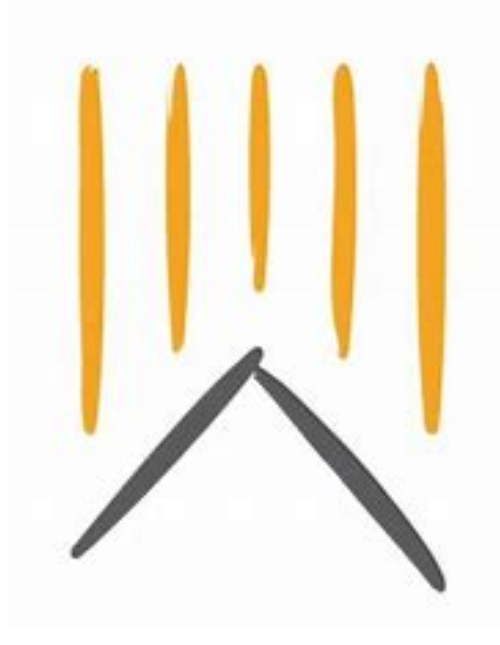
- Visualisation des textes : TEIPublisher
- Frise chronologique
- Intégration sur le serveur italien

Mise en œuvre

1. Création d'un odd
2. Création d'un html et d'un javascript
3. Actuellement sur serveur local

Résultats attendus

Voici la frise chronologique finale :



Entrepôt des données XML et visualisations de concepts sémantiques

Résumé

Voici le résumé de ce PRD. Voici le résumé de ce PRD. Voici le résumé de ce PRD. Voici le résumé de ce PRD. Voici le résumé de ce PRD. Voici le résumé de ce PRD. Voici le résumé de ce PRD. Voici le résumé de ce PRD.

Mots-clés

motcle1, motcle2, etc.

Abstract

Here is the abstract of this project. Here is the abstract of this project. Here is the abstract of this project. Here is the abstract of this project. Here is the abstract of this project. Here is the abstract of this project. Here is the abstract of this project. Here is the abstract of this project.

Keywords

word1, word2, etc.

Entreprise

Polytech



Tuteur entreprise

Giorgia GALLUCI

Étudiant

Clément BOUTONNET (DI5)

Tuteurs académiques

Mohamed SLIMANE

Elena PIERAZZO