

Stage PPE



Optimisation et Modernisation de la Génération du Programme Pédagogique

MACHKOUR Ayoub - TERRO Yassine

Période : 9 Septembre 2024 - 20
décembre 2024

Tuteur académique : CHOQUEUSE
Vincent

Tuteur entreprise : MENORET Mathilde

Remerciements

Au terme de ce projet PPE réalisé en binôme à l'ENIB, nous tenons à exprimer notre gratitude à toutes les personnes ayant contribué à sa réussite.

Nous adressons nos remerciements à **M. Pascal REDOU**, directeur des études, pour nous avoir permis de mener ce projet en remplacement du stage d'assistant ingénieur.

Nos sincères remerciements vont également à notre tuteur académique, **M. Vincent CHOQUEUSE**, et à notre tutrice en entreprise, **Mme Mathilde MENORET**, pour leur encadrement bienveillant, leurs précieux conseils et leur disponibilité constante. Leur accompagnement a été déterminant dans l'accomplissement de notre travail.

Enfin, nous tenons à remercier l'ensemble du personnel de l'ENIB, et plus particulièrement **Mme Chantal CALVES**, responsable des relations entreprises, pour sa disponibilité et les informations précieuses qu'elle nous a fournies.

Table des Matières

Remerciements	2
1 L'ENIB : Une école d'ingénieurs en évolution:	9
1.1 Engagements écologiques et développement durable	9
1.2 Infrastructures et équipements	10
1.3 Organisation et gouvernance	11
1.4 Formation et spécialités	12
1.5 Réseau des anciens élèves et insertion professionnelle	12
1.6 Les acteurs impliqués dans la données administratives	13
1.7 Enjeux et perspectives - Conclusion	14
2 Analyse du système existant	17
2.1 Système de Gestion de Contenu Actuel	17
2.2 Spécifications du Nouveau Système	22
3 Notre choix	26
3.1 Analyse comparative des alternatives	29
4 Comprendre Typst : Origines et Fonctionnalités	35
4.1 La Naissance de Typst	35
4.1.1 Contexte Historique	35
4.1.2 Évolution du Projet	35
4.2 Contexte et Point de Départ	36
4.3 Première Phase : Adaptation Initiale vers Typst	36
4.4 Adaptation des Templates et du Système de Filtrage	37
4.5 Structuration des Données et Organisation du Projet	39
4.6 Centralisation des Données et Automatisation	39
4.7 Optimisation et Contrôle de la Génération	40
4.8 Performances et Résultats	42
4.9 Illustration du Processus de Génération des Documents PDF avec Typst	42
4.10 Développement d'un Template de Rapport de Stage	43
4.10.1 Architecture du Template	43
4.10.2 Le Fichier Principal (main.typ)	43
4.11 Conclusion	44
5 Gestion des métadonnées YAML	46
5.1 Structure et organisation du code	46
5.1.1 Choix de la formation et modes d'édition	48
5.1.2 Modification individuelle	49
5.1.3 Opérations en lot	50
5.2 Interface Web avec Flask	52
5.3 Intégration FISEA dans VitePress	57
5.3.1 Contexte et état initial	57
5.3.2 Fusion des projets et objectifs FISEA	58
5.3.3 Mise à jour des scripts JSON et CSV	59
5.3.4 Adaptation de la configuration VitePress	60
5.3.5 Bénéfices de l'intégration	61
6 Gouvernance et organisation responsable	64

6.1 Impact environnemental et sobriété numérique	65
6.2 Relations de travail et inclusion numérique	65
6.3 Impact sociétal et développement numérique local	66
6.4 Comparaison avec d'autres établissements	67
6.5 Impacts indirects et effets numériques induits	67
6.6 Perspectives d'amélioration RSE-RNE	68
6.7 Recommandations pour une évolution durable	68
6.8 Conclusion	69
7 Conclusion générale :	71
BIBLIOGRAPHIE	72
ANNEXES	74

Table des Figures

Figure 1: Organisation de l'ENIB - Répartition des Pôles et Directions	11
Figure 2: Structure des métadonnées du cours (En-tête YAML)	18
Figure 3: Contenu pédagogique et acquis d'apprentissage du cours	19
Figure 4: Architecture technique du système documentaire (VitePress)	20
Figure 5: Architecture global du système de génération de PDF (LaTeX)	21
Figure 6: Commande tree pour le premier projet (LaTeX)	36
Figure 7: Commande tree pour le nouveau projet (Typst)	37
Figure 8: Exemple de tableau en LaTeX	38
Figure 9: Exemple de tableau en Typst	39
Figure 10: Génération Centralisée des Fichiers JSON(FISEA/FISEA)	40
Figure 11: fichier de configuration YAML	41
Figure 12: Diagramme de séquence- génération de documents PDF avec Typst	42
Figure 13: Commande tree pour le template de rapport	43
Figure 14: Résultat de la Commande 'tree' : Structure des Répertoires	47
Figure 15: Diagramme de classes du Système de Gestion des Métadonnées YAML	47
Figure 16: Interface de sélection du mode de modification.	48
Figure 17: Structure et navigation de l'outil de gestion des métadonnées	49
Figure 18: Diagramme d'État – Mode Individuel de Modification des Fichiers YAML	50
Figure 19: Menu des opérations par lot	51
Figure 20: Diagramme d'État – Processus de Modification en Mode Groupé	51
Figure 21: Diagramme de Cas d'Utilisation – Gestion et Génération de Documents PDF	53
Figure 22: Vue d'ensemble du tableau de bord avec l'état des fichiers générés	54
Figure 23: Configuration de la page de garde	54
Figure 24: Configuration des sections de contenu du document	55
Figure 25: Paramètres d'affichage des informations de cours	55
Figure 26: Interface de sélection des cours et gestion des volumes horaires	56
Figure 27: Pipeline de Génération VitePress et PDF (Avant Intégration FISEA)	58
Figure 28: Pipeline de Génération VitePress et PDF (Après Intégration FISEA)	59
Figure 29: Interface initiale VitePress affichant uniquement les semestres FISE	60
Figure 30: Nouvelle interface VitePress intégrant les semestres FISE et FISEA	61

Liste des Tableaux

Table 1: Outils et Technologies du Projet	28
Table 2: Python et ses alternatives	29
Table 3: Miniconda et ses alternatives	29
Table 4: Flask et ses alternatives	30
Table 5: Jinja2 et ses alternatives	30
Table 6: Vitepress et ses alternatives	31
Table 7: Make et ses alternatives	31
Table 8: Pandoc et ses alternatives	32
Table 9: Typst et ses alternatives	32

Introduction Générale

L'**École Nationale d'Ingénieurs de Brest (ENIB)**, récemment transformée en établissement public à caractère scientifique, culturel et professionnel (**EPSCP**), s'apprête à intégrer en 2025 un nouvel Institut National Polytechnique (INP Bretagne) aux côtés de l'**ESIAB** et de l'IAE. Dans ce contexte de transformation institutionnelle, l'ENIB enrichit également son offre de formation avec l'ouverture d'une nouvelle voie de **formation d'ingénieurs par apprentissage (FISEA)**, accréditée par la **Commission des Titres d'Ingénieur (CTI)**.

Cette évolution s'accompagne d'un besoin croissant de gestion documentaire sophistiquée, notamment pour répondre aux exigences de différents acteurs. La CTI, en tant qu'organisme d'accréditation, requiert une documentation précise et structurée du programme pédagogique. **Les Opérateurs de Compétences (OPCO)**, qui financent la formation des étudiants en alternance, exigent quant à eux des programmes de formation personnalisés. Le service des relations internationales nécessite également des documents adaptés pour les mobilités étudiantes.

Jusqu'à présent, la préparation de ces documents reposait sur un processus manuel chronophage, impliquant de multiples manipulations de fichiers et sources de données, source potentielle d'erreurs et d'incohérences. Notre projet vise à répondre à ces enjeux en développant un outil permettant la génération automatisée et personnalisée des programmes de formation, tout en garantissant une source unique de vérité pour l'ensemble des données pédagogiques.

Ce rapport s'articule autour de 6 chapitres qui retracent notre démarche et nos réalisations. **Le chapitre 1** présente le contexte de transformation de l'école et les enjeux documentaires associés. **Le chapitre 2** dresse un état des lieux du système de gestion documentaire actuellement en place à l'ENIB et décrit les besoins fonctionnels du projet. **Le chapitre 3** présente et détaille les outils et technologies retenus pour répondre aux besoins, en justifiant ces choix. **Le chapitre 4** présente l'implémentation de cet outil moderne pour la génération automatisée des documents PDF ainsi que la réalisation de ce rapport. **Le chapitre 5** détaille les améliorations apportées au site existant et le développement d'une nouvelle interface web pour personnaliser les documents PDF. **Le chapitre 6** analyse les aspects environnementaux et sociétaux de la solution.

Chapitre 1: Contexte institutionnel et stratégique de l'ENIB

Introduction

Ce premier chapitre présente l'École Nationale d'Ingénieurs de Brest (ENIB), établissement au sein duquel notre projet a été développé. Cette présentation permettra de comprendre le contexte institutionnel, les spécificités de l'école et les enjeux liés à la gestion de ses documents pédagogiques. Nous aborderons notamment son organisation, ses formations et sa collaboration avec les différents acteurs du monde professionnel.

1 L'ENIB : Une école d'ingénieurs en évolution:

Crée en 1961, l'**École Nationale d'Ingénieurs de Brest** est l'une des 204 écoles d'ingénieurs françaises accréditées[1]. Située dans le Technopôle Brest-Iroise à Plouzané, l'ENIB a récemment évolué vers le statut d'établissement public à caractère scientifique, culturel et professionnel (**EPSCP**). Cette évolution marque une étape importante dans le développement de l'établissement. Au 1er janvier 2025, l'ENIB franchira une nouvelle étape en devenant une école interne d'un **Institut National Polytechnique (INP)**, lui-même membre de l'établissement public expérimental de l'université de Bretagne occidentale. Cette transformation s'inscrit dans une stratégie de renforcement des synergies locales, l'INP accueillant également l'ESIAB et un **Institut d'Administration des Entreprises (IAE)**.

1.1 Engagements écologiques et développement durable

L'ENIB place le développement durable et la responsabilité sociétale au cœur de sa mission éducative. Depuis près de quinze ans, l'école intègre les transitions socio-écologiques et les questions éthiques dans ses programmes d'enseignement. Tous les étudiants et le personnel sont formés et sensibilisés aux enjeux de la transition écologique et du développement soutenable[2].

En 2023, l'ENIB a introduit le module « Projet éco-responsable », consolidant ainsi son engagement envers une formation centrée sur les **Objectifs de Développement Durable (ODD)**. L'école collabore également avec des institutions locales, telles que le campus brestois de l'**École européenne supérieure d'art de Bretagne (EESAB)**, pour aborder le design de la transition à travers des activités pédagogiques conjointes et offrir des parcours doubles aux élèves-ingénieurs.

Par ailleurs, l'ENIB a entrepris des actions concrètes pour réduire son empreinte carbone. En 2022-2023, l'école a rénové son système d'éclairage en adoptant la technologie LED dans l'ensemble de ses bâtiments, une initiative financée en partie par le **Fonds européen de développement régional (FEDER)**. Cette modernisation vise à diminuer de 23% la consommation électrique de l'établissement.

1.2 Infrastructures et équipements

L'ENIB met à disposition des infrastructures modernes et diversifiées pour soutenir la formation et la recherche de ses étudiants et chercheurs. Ces équipements favorisent l'innovation, la pratique et la réalisation de projets concrets.

- **Centre Européen de Réalité Virtuelle (CERV)** : Créé en 2004, le CERV est une plateforme dédiée à la recherche et à l'innovation en réalité virtuelle. Équipé de systèmes immersifs et interactifs, il permet de mener des projets en modélisation et simulation. Le CERV accueille des chercheurs du Lab-STICC (UMR6285 CNRS) et d'autres partenaires académiques et industriels. Les étudiants participent également à des projets, renforçant leurs compétences dans des domaines comme l'industrie, la santé ou la défense[3].
- **La Forge** : Inaugurée en 2023, La Forge est un espace de prototypage conçu pour permettre aux étudiants de concrétiser leurs projets d'ingénierie. Équipée d'imprimantes 3D, de fraiseuses numériques et d'outils de conception assistée par ordinateur, elle encourage la créativité, l'innovation et l'apprentissage par la pratique dans des domaines tels que la mécanique, la robotique et l'électronique[4].

Laboratoires Partenaires

L'ENIB s'appuie sur des laboratoires de recherche partenaires pour offrir un accès à des équipements de haute technologie.

- **Institut de Recherche Dupuy de Lôme (IRDL)**: Spécialisé en ingénierie mécanique et science des matériaux, l'IRDL mène des études sur la durabilité des structures et la fatigue des matériaux. Il dispose de machines d'essais avancées et de systèmes de mesure permettant d'analyser les chocs et vibrations, contribuant ainsi à des projets dans les secteurs maritimes, aéronautique et automobile[5].
- **Lab-STICC (UMR CNRS 6285)**: Laboratoire multidisciplinaire, le Lab-STICC travaille sur l'électronique, la photonique, les hyperfréquences et les télécommunications. L'ENIB participe activement aux projets de recherche, permettant aux étudiants de se former sur des équipements de pointe dans les domaines des systèmes embarqués, du traitement du signal et des télécommunications optiques[6].

Projets Complémentaires

- **Lab'Optic:** Créé en 2024, ce laboratoire commun avec IMT Atlantique, l'ENSSAT et Orange se concentre sur les télécommunications optiques. Il vise à développer des technologies pour améliorer les réseaux à haut débit, impliquant les étudiants dans des projets innovants.
- **Subventions et Nouveaux Équipements :** L'ENIB bénéficie de financements européens et régionaux pour l'acquisition d'équipements de recherche et de prototypage. Ces investissements renforcent la formation pratique des étudiants et leur permettent de travailler sur des outils industriels de dernière génération.

Grâce à ses infrastructures internes et à ses collaborations avec des laboratoires partenaires, l'ENIB garantit à ses étudiants un environnement d'apprentissage riche, favorisant l'acquisition de compétences techniques et la réalisation de projets innovants.

1.3 Organisation et gouvernance

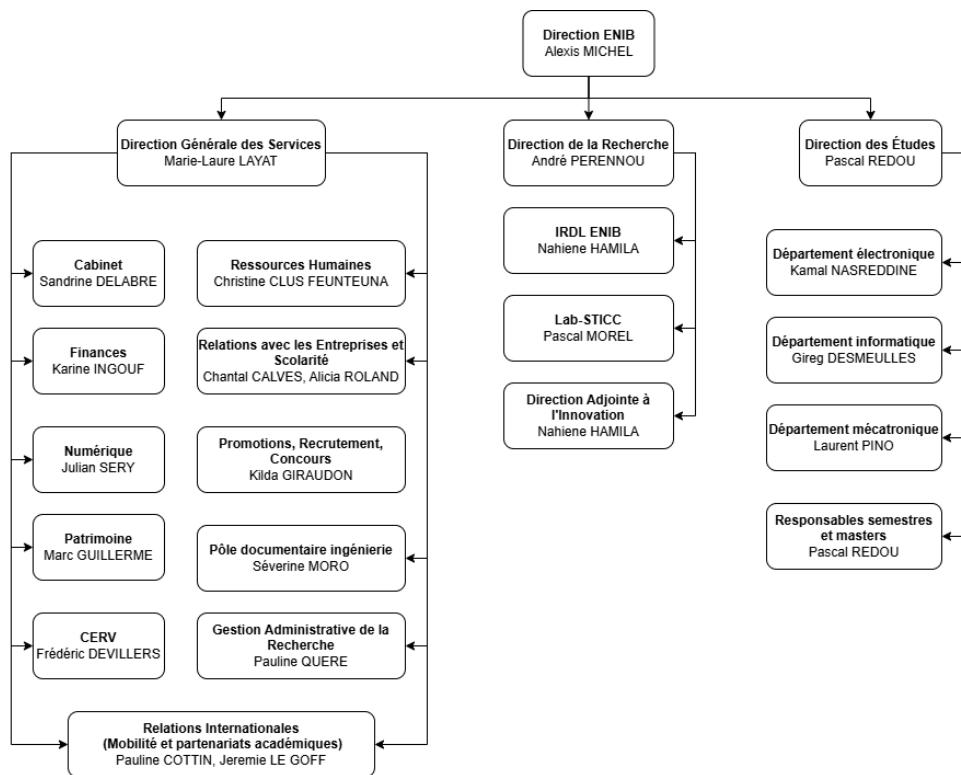


Figure 1: Organisation de l'ENIB - Répartition des Pôles et Directions

L'organisation de l'ENIB repose sur trois pôles principaux, comme illustré dans l'organigramme ci-dessus :

- **Le pôle Études** Il regroupe trois départements d'enseignement : électronique, informatique et mécatronique, qui assurent la formation des étudiants dans les disciplines clés de l'ingénierie.
- **Le pôle Recherche**, supervise les relations internationales et les relations entreprises, renforçant ainsi les collaborations scientifiques et industrielles. Il encadre également les activités de recherche menées dans les laboratoires partenaires.
- **La Direction Générale des Services (DGS)**, chargée de la gestion administrative et des ressources. Elle s'occupe des ressources humaines, des finances, de la communication et des services liés à la scolarité et aux relations entreprises.

L'école emploie 126 personnes, comprenant 73 enseignants (dont 53 permanents à l'ENIB) et 53 personnels BIATSS (Bibliothèques, Ingénieurs, Administratifs, Techniciens, de Santé et de Service). Cette organisation garantit un encadrement optimal pour les 768 élèves actuellement en formation, avec une part significative d'internationaux (21%) et de femmes (18%).

1.4 Formation et spécialités

L'ENIB délivre un seul diplôme d'ingénieur généraliste avec trois dominantes : électronique, informatique et mécatronique[7]. La Commission des Titres d'Ingénieur (CTI) reconnaît trois voies traditionnelles d'obtention du diplôme d'ingénieur : la Formation Initiale sous Statut Étudiant (FISE), la Formation Initiale sous Statut Apprenti (FISA), et la Formation Continue (FC). Actuellement, l'ENIB propose ce diplôme principalement par la voie FISE, qui constitue un cursus de 5 ans basé sur un tronc commun équilibré et le contrôle continu, ainsi que par la Formation Continue avec possibilité de Validation des Acquis de l'Expérience (VAE)[8].

La nouvelle voie FISEA, en cours de mise en place, vient enrichir cette offre avec un parcours hybride innovant combinant une première année sous statut étudiant suivie de deux années sous statut apprenti[9]. Cette formation, qui fait partie des nouvelles modalités d'accès au diplôme d'ingénieur reconnues par la CTI depuis 2019, accueillera 24 étudiants par promotion, dont 14 recrutés à Bac+2 et 10 issus du cycle préparatoire de l'ENIB. Cette voie se distingue de la FISA traditionnelle par son format mixte étudiant-apprenti. En 2024, 31 formations d'ingénieurs en France étaient accréditées par la CTI pour proposer ce type de formation mixte, illustrant l'essor de cette approche pédagogique innovante et la volonté des écoles de diversifier leurs modalités d'apprentissage[10].

1.5 Réseau des anciens élèves et insertion professionnelle

Depuis sa création, l'ENIB a formé plus de 5 000 ingénieurs, répartis en France et dans plus de 60 pays[7]. L'Association Nationale des Ingénieurs de l'ENI de Brest (ANIENIB) joue un

rôle clé dans l'animation de ce réseau [11], en organisant des événements tels que des tables rondes métiers et des simulations d'entretien d'embauche.

Ces initiatives permettent aux étudiants de bénéficier de l'expérience des anciens, d'appréhender la diversité des métiers d'ingénieur et de se préparer efficacement à leur insertion professionnelle. L'ENIB met également l'accent sur l'interaction avec les entreprises dès le début du cursus, facilitant ainsi l'accès à des opportunités de carrière variées dans des secteurs d'activité porteurs[12].

En complément, l'école organise régulièrement des journées entreprises, des entretiens professionnels et des conférences techniques, renforçant les liens entre les étudiants et le monde professionnel. Cette approche proactive vise à assurer une transition fluide des diplômés vers le marché du travail, en adéquation avec les besoins des industries contemporaines[12].

1.6 Les acteurs impliqués dans la données administratives

Les organisations externes et leurs exigences:

La Commission des Titres d'Ingénieur (CTI) est l'organisme indépendant chargé, depuis 1934, d'évaluer et d'accréditer les formations d'ingénieurs en France. L'ENIB doit régulièrement fournir des documents précis et conformes à ses exigences lors des audits périodiques ou sur demande spécifique. Ces exigences documentaires sont détaillées dans le Référentiel 2024 de la CTI [13] et comprennent plusieurs catégories essentielles.

Les documents stratégiques et administratifs incluent les statuts de l'école, les contrats d'objectifs et la note stratégique. Les documents pédagogiques comprennent le syllabus complet, le règlement des études, les maquettes de diplômes, une vue d'ensemble des semestres et le programme pédagogique détaillé. En ce qui concerne les documents de suivi et résultats, l'école doit fournir les données certifiées sur les effectifs ainsi que les enquêtes d'insertion professionnelle. Pour la vie étudiante, les documents requis sont notamment les chartes associatives et le livret d'accueil. Tous ces documents doivent être regroupés dans un dossier numérique accessible aux auditeurs et régulièrement mis à jour par l'école pour répondre aux exigences d'évaluation et d'accréditation de la CTI.

Les Opérateurs de Compétences (OPCO), créés en 2019, sont des organismes étatiques essentiels dans l'écosystème de la formation professionnelle[14]. Ils supervisent le financement et le suivi des formations en alternance à l'ENIB, tant pour les contrats de professionnalisation que pour les contrats d'apprentissage[15]. Pour assurer ce suivi, les OPCO requièrent une documentation détaillée et personnalisée pour chaque étudiant, spécifiant précisément les modules de formation suivis. Notre outil répond à ce besoin en générant automatiquement des documents adaptés au parcours individuel de chaque étudiant.

Les structures internes de l'ENIB:

La gestion documentaire à l'ENIB repose principalement sur l'interaction entre deux services clés. Le Service Relations Entreprises et Scolarité, sous la responsabilité de Madame Chantal CALVES, est en charge de la production et de la validation des documents pédagogiques. Son équipe assure quotidiennement la gestion administrative des formations, la génération des conventions de stage et des contrats d'alternance, ainsi que le suivi des documents liés aux parcours des étudiants.

Le Service Numérique (SNUM), dirigé par Julian SERY, apporte le support technique nécessaire à cette gestion documentaire. Il supervise l'infrastructure informatique et assure la sécurité des données. L'équipe technique du SNUM maintient les systèmes d'information existants et accompagne l'intégration de nouvelles solutions, comme notre outil de génération automatisée de documents pédagogiques.

Cette organisation permet une gestion efficace des documents pédagogiques : les aspects métier sont pilotés par le service Relations Entreprises et Scolarité qui définit les besoins et valide les contenus, tandis que le SNUM assure l'intégration technique et le support des outils mis en place. Cette collaboration étroite entre les deux services est essentielle pour garantir une gestion documentaire fiable et efficace, répondant aux exigences de tous les acteurs de la formation.

1.7 Enjeux et perspectives - Conclusion

L'évolution de l'ENIB, tant sur le plan institutionnel avec sa transformation en **EPSCP** et son intégration prochaine à l'**INP**, que sur le plan pédagogique avec l'introduction de la voie **FISEA**, génère des besoins croissants en matière de gestion documentaire. Cette évolution s'accompagne de nombreux défis dans la production et la gestion des documents pédagogiques.

En premier lieu, l'école doit répondre aux exigences documentaires de différents acteurs. Les documents doivent être adaptés aux besoins spécifiques de la CTI pour les accréditations, des **OPCO** pour le suivi des alternants, et du service des relations internationales pour la mobilité des étudiants. Cette diversité des destinataires nécessite une grande flexibilité dans la production des documents, tout en maintenant leur cohérence.

Par ailleurs, la multiplication des parcours de formation (voie classique, contrats de professionnalisation et apprentissage) complexifie la gestion documentaire. Les documents pédagogiques doivent pouvoir s'adapter aux différents cursus tout en garantissant une cohérence globale de la formation.

Face à ces enjeux, le développement d'un outil adapté et évolutif pour la gestion documentaire devient indispensable. Cette compréhension approfondie du contexte nous conduit naturellement au chapitre suivant, qui détaillera le cahier des charges de notre projet.

CHAPITRE 2 : Analyse du système existant et Spécifications

Introduction

Ce chapitre présente une analyse détaillée du système de gestion documentaire actuellement en place à l'ENIB. Mis en œuvre en mars 2023, ce système repose sur une architecture combinant un site web statique généré par VitePress et un processus de génération de documents PDF via LaTeX. Cette analyse permet de comprendre l'organisation actuelle, ses points forts et ses limitations, éléments essentiels pour appréhender les évolutions nécessaires face aux nouveaux besoins de l'école.

2 Analyse du système existant

2.1 Système de Gestion de Contenu Actuel

Évolution et contexte

Le système documentaire de l'ENIB a été profondément remanié en mars 2023 pour répondre aux nouvelles exigences pédagogiques et institutionnelles. Initialement, le programme pédagogique était élaboré en LaTeX, avec une production directe de documents PDF. Bien que cette méthode garantisse une mise en page professionnelle, elle s'est révélée inadaptée face à des besoins pédagogiques croissants. En 2021, la Commission des Titres d'Ingénieur (CTI) a souligné la nécessité d'intégrer l'approche par compétences (APC)[16], imposant une restructuration majeure du système documentaire pour mieux répondre aux standards et attentes de l'évaluation académique.

Pour accompagner cette transformation, l'ENIB a recruté Mme Mathilde MENORET ingénierie pédagogique, afin de garantir la cohérence et l'efficacité dans la mise en œuvre des nouvelles solutions. Ce recrutement stratégique s'est accompagné de l'adoption de VitePress en mars 2023. Ce framework moderne permet de générer automatiquement un site web statique à partir de fichiers Markdown, facilitant ainsi la diffusion et l'accessibilité des contenus pédagogiques. Le système a été enrichi par l'intégration de GitLab, qui offre une plateforme centralisée pour stocker, versionner et modifier les fichiers Markdown, tout en assurant un contrôle précis des contributions des enseignants.

Structure et Organisation du Contenu

Le contenu pédagogique est désormais structuré en deux parties principales dans chaque fichier Markdown.

- **La première partie** comprend un en-tête YAML qui contient les métadonnées essentielles du cours : les acquis d'apprentissage terminaux (AAT), le coefficient, l'identifiant unique du cours, ainsi que les volumes horaires détaillés pour chaque type d'enseignement (cours magistraux, TD, TP, etc.).

```

---
aat:
- E1
- F2
coeff: '2'
ds: '0'
id: 01_XFITA
optional: 'true'
supervisor: Not Assigned
location: enib
type_cti: langue
ue: F
module: ''
name: ''
volume:
  cm: '0'
  ctd_des: '0'
  ctd_enc: '0'
  labo_des: '0'
  labo_enc: '21'
  perso: '19'
  projet_encadre: '0'
  projet_perso: '0'
  td_des: '0'
  td_enc: '0'
  total_edt: '21.0'
  total_encadre: '21.0'
  total_encadre_perso: '40.0'
  tp_des: '0'
  tp_enc: '0'
tags:
- name: Recherche
  ratio: 1.0
- name: Innovation
  ratio: 1.0
- name: Rse
  ratio: 1.0
---

```

Figure 2: Structure des métadonnées du cours (En-tête YAML)

- **La seconde partie** présente le contenu pédagogique proprement dit, organisé en sections distinctes : la description générale du cours, les acquis d'apprentissage visés (AAV) avec leur volume horaire et les compétences associées, les modalités d'évaluation, les mots-clés, les prérequis et les ressources pédagogiques. Cette structure permet une présentation claire et cohérente des informations pédagogiques.

```

# Italien (01_XFITA)

<Course />

## Description
Italien niveau débutant
## Acquis d'Apprentissage visés (AAv)

* AAv1_deb [heures:10,F2]: A l'issue du cours, l'étudiant.e sera capable de s'exprimer en italien par de courtes phrases pour donner des informations factuelles sur son identité et sur des sujets de la vie quotidienne à l'oral et à l'écrit.

* AAv2_deb [heures:10,E1]: A l'issue du cours, l'étudiant.e sera capable d'identifier le vocabulaire en italien lié aux projets et aux activités et de reproduire et d'adapter les structures observées.

* AAv3_deb [heures:10,F2]: A l'issue du cours, l'étudiant.e sera capable de construire en italien des énoncés simples présent et au futur proche, en conjuguant correctement les verbes et en utilisant le vocabulaire à sa disposition, si besoin à l'aide d'outils numériques d'aide linguistique.

## Modalités d'évaluation
Moyenne de plusieurs épreuves de contrôle continu courtes

## Mots clés

## Pré-requis

## Ressources
  
```

Figure 3: Contenu pédagogique et acquis d'apprentissage du cours

Processus de Gestion et Organisation du Système

Le système s'appuie sur un dépôt GitLab centralisé qui structure la gestion et la validation des contenus pédagogiques. Cette plateforme accueille 112 contributeurs, majoritairement des enseignants disposant de droits de développeur. La supervision est assurée par trois administrateurs clés : M. Vincent CHOQUEUSE, Mme Mathilde MENORET et M. Gilles CHAUVEAU.

L'architecture du système s'articule autour de trois composants principaux : les fichiers sources, les outils de traitement Python, et les fichiers générés. Les fichiers sources comprennent les fichiers de configuration, les fichiers JSON définissant la structure des cours et des AAT, ainsi que les fichiers de contenu incluant les descriptions des cours et des templates LaTeX. Les outils Python (build_json.py, build_csv.py, build_md.py et build_latex.py) traitent ces données sources pour générer différents types de fichiers : des fichiers CSV documentant les volumes horaires et les mappings UE-AAT-AAV, des fichiers JSON structurant la hiérarchie AAT-AAV, des fichiers Markdown pour la documentation, et des fichiers LaTeX pour la production de documents PDF.

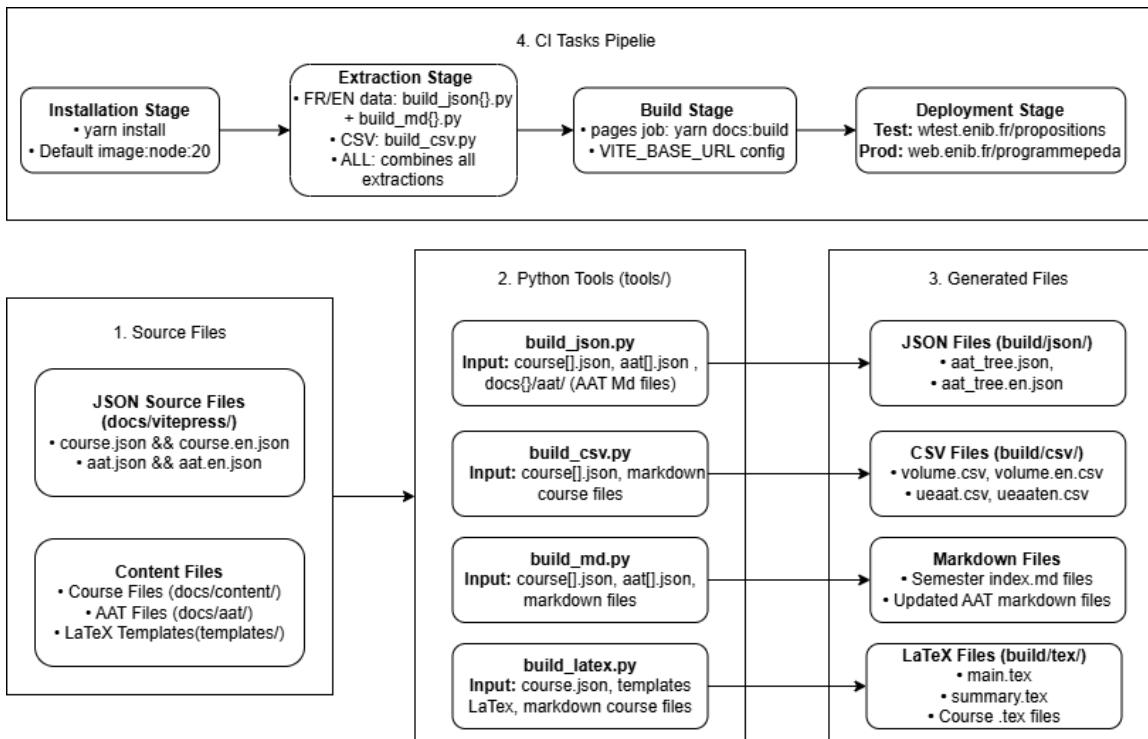


Figure 4: Architecture technique du système documentaire (VitPress)

Le workflow de validation assure un contrôle qualité rigoureux : chaque modification est systématiquement soumise à l'approbation de l'ingénieur pédagogique avant publication. L'intégration continue est gérée par GitLab CI/CD à travers un pipeline comprenant plusieurs étapes automatisées. L'étape d'installation configure l'environnement Node.js et installe les dépendances nécessaires. L'étape d'extraction, déclenchée manuellement, exécute les scripts Python pour générer les fichiers JSON, Markdown et CSV, avec des versions distinctes pour le français et l'anglais. La phase de construction utilise VitePress pour générer le site statique. Enfin, l'étape de déploiement permet de publier le site soit dans un environnement de test accessible via www.wtest.enib.fr/propositions, soit en production sur web.enib.fr/programmepeda. Cette automatisation garantit une génération cohérente et systématique de l'ensemble des documents du programme pédagogique.

Processus de génération documentaire:

Parallèlement au site web, un système de génération de documents PDF a été développé spécifiquement pour l'accréditation de la formation FISEA en 2024. Ce processus utilise une chaîne d'outils permettant de produire le programme pédagogique complet de la formation au format PDF. Python assure le traitement des données et l'automatisation générale du processus. Jinja2 est utilisé pour générer des fichiers LaTeX à partir de templates, en les alimentant avec des données structurées en Python. Pandoc permet la conversion des fichiers Markdown vers le format LaTeX, tandis que des filtres Lua personnalisés assurent

le traitement des balises spécifiques comme les informations de cours et les tableaux de volumes horaires. LaTeX finalise la chaîne en assurant la mise en forme professionnelle des documents.

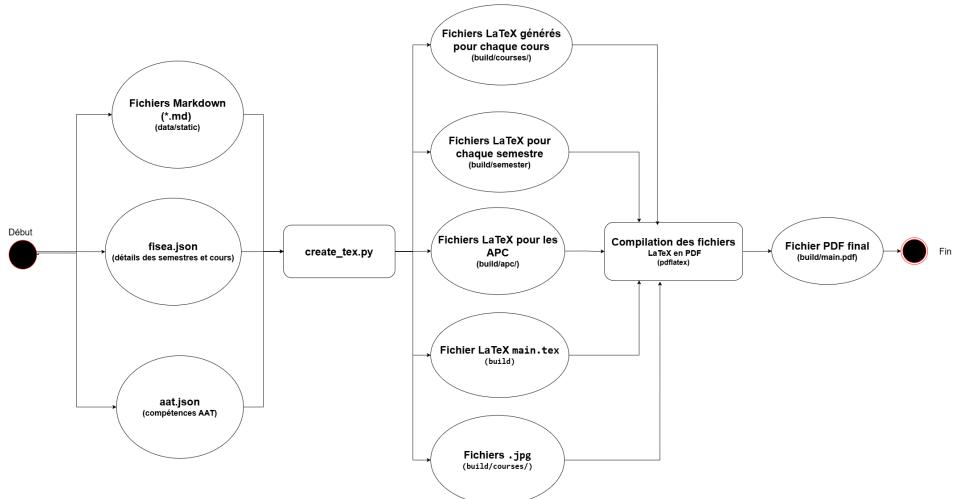


Figure 5: Architecture global du système de génération de PDF (LaTeX)

Cette chaîne de traitement permet de générer des documents PDF structurés à partir des contenus Markdown, en suivant une séquence précise de transformations et de mise en forme.

Limitations et défis actuels

La gestion documentaire actuelle de l'ENIB repose sur deux systèmes parallèles : un site web et la génération de PDF. Bien que fonctionnels, ces modes de fonctionnement présentent des limites en matière d'optimisation et d'efficacité.

Le site web, reconnu pour sa qualité et sa capacité à mettre en valeur la formation FISE, ne prend pas encore en compte la nouvelle formation FISEA. Parallèlement, la génération automatisée de PDF, déjà opérationnelle pour la FISEA, n'a pas été étendue à la formation FISE, ce qui crée un déséquilibre entre les deux approches.

Un défi majeur réside dans la production de documents personnalisés pour les organismes externes. Les demandes spécifiques, comme celles de l'OPCO pour les contrats de professionnalisation ou de la CTI pour les évaluations, nécessitent un investissement considérable en temps de la part du personnel administratif. Actuellement, l'extraction d'informations depuis le site web ou le partage de liens permet de répondre à ces besoins, mais au prix d'une mobilisation importante des ressources humaines. Une automatisation plus poussée permettrait de libérer du temps pour des tâches à plus forte valeur ajoutée, tout en maintenant le niveau d'excellence reconnu de l'établissement.

Le processus de génération de PDF basé sur LaTeX, bien qu'efficace, présente plusieurs limites techniques :

- La complexité de son utilisation requiert une expertise spécifique.
- Les temps de compilation sont relativement longs.
- La maintenance des templates s'avère laborieuse.
- Les modifications rapides de mise en forme représentent un défi important.

2.2 Spécifications du Nouveau Système

Objectifs et Vision

Face aux nouveaux besoins liés à l'introduction de la formation FISEA et aux limitations du système de génération PDF existant, l'ENIB a entrepris une évolution majeure de sa gestion documentaire. Cette évolution repose sur deux axes complémentaires et techniquement indépendants.

Extension du Site Web VitePress

L'extension du site VitePress permet d'unifier la présentation des formations FISE et FISEA au sein de la même plateforme web, garantissant une navigation fluide et cohérente. Les scripts existants ont été modifiés pour gérer les deux formations de manière harmonisée. La formation FISEA a été intégrée en réutilisant la structure de la FISE, évitant ainsi les duplications inutiles tout en assurant une cohérence globale.

Afin de répondre aux exigences d'internationalisation, des fiches de cours en anglais ont été créées pour la formation FISEA. Cette démarche assure une présentation bilingue, déjà appliquée à la FISE, et répond ainsi aux attentes des partenaires et des étudiants étrangers. L'ensemble du site continue de s'appuyer sur des fichiers Markdown, simplifiant la rédaction et la mise à jour des contenus par les enseignants et administrateurs, sans nécessiter de compétences techniques particulières.

Nouveau Système de Génération PDF

La refonte du système de génération PDF repose sur le remplacement de LaTeX par Typst, une solution moderne qui améliore significativement la rapidité de compilation et simplifie la maintenance. Avec Typst, les PDF sont générés plus rapidement tout en respectant la charte graphique de l'ENIB. Cette transition réduit la complexité associée à LaTeX, rendant le système plus stable et fiable.

Une interface web développée avec Flask accompagne cette nouvelle architecture. Grâce à cette interface, les utilisateurs peuvent configurer et générer des documents PDF sans avoir à manipuler directement des scripts ou des fichiers de configuration complexes. La sélection

des semestres, des cours et l'ajustement des volumes horaires s'effectuent désormais via une interface intuitive accessible depuis un navigateur web.

Gestion des Métadonnées

La gestion des métadonnées a été optimisée grâce au développement d'un script Python. Cet outil permet de modifier les en-têtes YAML des fichiers Markdown directement depuis la ligne de commande. Il offre plusieurs fonctionnalités essentielles, telles que l'ajout de nouveaux champs, la modification des informations existantes et la suppression de champs obsolètes. Les administrateurs peuvent appliquer ces changements à un fichier ou à l'ensemble des fiches de cours, garantissant une cohérence des métadonnées entre les formations FISE et FSEA.

Automatisation et Configuration

L'utilisation du système a été simplifiée par la création d'un Makefile qui automatise les tâches courantes. Les commandes permettent de générer les PDF complets ou de compiler uniquement les fichiers Typst nécessaires. La personnalisation des documents repose sur un fichier de configuration YAML centralisé, qui permet d'ajuster la table des matières, les volumes horaires et la mise en page des documents finaux. La configuration de la page de garde, des sections à inclure ou à exclure, ainsi que des tableaux et graphiques, est également gérée à travers ce fichier YAML.

Cette approche modulaire prévient les erreurs liées aux modifications directes des templates et scripts, tout en accélérant les ajustements en fonction des demandes de partenaires internes ou externes de l'école.

Interface Web et Gestion des Versions

L'interface web Flask joue un rôle clé dans la gestion documentaire en remplaçant la modification manuelle des fichiers YAML par une solution graphique et intuitive. Cette interface permet aux utilisateurs de personnaliser les cours, de sélectionner les semestres et d'ajuster les volumes horaires sans avoir à écrire de code. Elle offre la génération automatisée de fichiers JSON, Typst et PDF, garantissant une production rapide et uniforme des documents.

L'interface permet également de basculer facilement entre les branches GitLab pour accéder aux versions précédentes des fiches de cours.

Gestion des Données

La génération des fichiers JSON repose sur le principe de "single source of truth". Les données pédagogiques sont consolidées dans deux fichiers JSON distincts, l'un pour la FISE et

l'autre pour la FISEA. Cette approche garantit que toutes les informations sont regroupées en un seul endroit, simplifiant ainsi la maintenance et évitant la dispersion des données à travers de multiples fichiers Markdown.

Architecture Unifiée

Même si les systèmes de génération PDF et de gestion du site web VitePress sont développés séparément, ils partagent une source commune, à savoir les fichiers Markdown stockés dans GitLab. Cette unification garantit que toute mise à jour effectuée sur le site web est immédiatement reflétée dans les documents PDF générés. L'utilisation de GitLab comme référentiel unique pour toutes les données pédagogiques assure la cohérence et la continuité des informations entre les différents supports.

Conclusion

L'analyse du système existant révèle une architecture documentaire robuste mais confrontée à des défis croissants. Le site web VitePress démontre son efficacité pour la FISE mais nécessite une extension pour intégrer la FISEA. Parallèlement, le système de génération PDF basé sur LaTeX, bien que fonctionnel, présente des limitations significatives en termes de performance et de facilité d'utilisation.

Face à ces constats, une évolution du système s'impose selon deux axes principaux : l'extension du site web pour une gestion unifiée de la FISE et de la FISEA, et le développement d'une nouvelle chaîne de génération PDF plus efficace. Cette transformation demande une analyse approfondie des différentes technologies disponibles, notamment pour la génération de documents PDF, l'automatisation des processus et le développement d'interfaces utilisateur. Le chapitre suivant présentera une étude comparative des solutions envisagées, justifiant chaque choix technique par rapport aux alternatives existantes, avec une attention particulière aux critères de performance, de maintenabilité et d'ergonomie qui ont guidé nos décisions.

CHAPITRE 3 : Choix techniques

Introduction

La génération automatisée de documents pédagogiques représente un défi technique majeur, nécessitant une architecture robuste capable de répondre à des besoins multiples : traitement efficace des données, production de documents de haute qualité et interface utilisateur intuitive. Dans ce contexte, le choix des technologies et outils constitue une étape cruciale qui impactera directement la performance, la maintenabilité et l'évolutivité du système. Ce chapitre présente l'architecture technique mise en place pour répondre aux exigences spécifiques de l'ENIB. Nous commencerons par détailler chaque composant technologique sélectionné, en expliquant son rôle précis dans l'architecture globale. Ensuite, nous analyserons de manière comparative les alternatives disponibles pour chaque technologie, en justifiant nos choix selon des critères objectifs tels que la performance, la facilité d'utilisation et la maturité des solutions. Une attention particulière sera portée à la cohérence de l'ensemble des outils sélectionnés, leur intégration mutuelle et leur capacité à former un écosystème technologique performant et pérenne.

3 Notre choix

Outil & Logo	Description	Utilisation
Python 3.12.6 ★53.4k (CPython) 	Langage de programmation de haut niveau, interprété, orienté objet, avec un typage dynamique fort	Au cœur du système, Python orchestre le processus complet de génération des documents, en gérant le traitement des fichiers JSON et Markdown ainsi que les métadonnées. Il assure le développement web via Flask pour l'interface utilisateur et le traitement des données. Python gère également l'automatisation avec des scripts de génération JSON et la synchronisation avec le système de versionnement, tout en assurant les calculs des volumes horaires et crédits ECTS.

Miniconda ★ 3.3k 	Distribution minimale de l'environnement Conda, gestionnaire de packages et d'environnements Python	Miniconda assure la gestion centralisée de l'environnement de développement et de production via l'environnement isolé typst_env. Il gère précisément les versions des bibliothèques clés comme Python 3.12.6, Typst 0.11.1 et Pandoc 3.5, garantissant une reproduction exacte de l'environnement sur différentes machines pour un comportement cohérent du système.
Flask 3.0.3 ★ 65.5k 	Framework web léger en Python pour la création d'applications web	Développement d'une interface web complète pour : <ol style="list-style-type: none"> 1. Affichage et modification de la configuration via config.yaml 2. Prévisualisation des documents générés 3. Gestion des uploads de fichiers Markdown
Jinja2 3.1.4 ★ 9.6k 	Moteur de templates Python pour la génération de documents	<ol style="list-style-type: none"> 1. Génération dynamique des fichiers Typst à partir des templates. 2. Transformation des données JSON et Markdown en documents structurés. 3. Permet la séparation entre la logique et la présentation des documents.
Pandoc 3.5 ★ 31.2k 	Convertisseur universel de documents supportant de nombreux formats	Pandoc assure la conversion des fichiers Markdown vers les formats intermédiaires via des filtres LUA personnalisés, gérant le traitement des métadonnées YAML et l'extraction des données de cours. Il s'intègre au système de templates pour la génération des tables des matières, la création des index et la gestion des références, tout en appliquant des filtres spécifiques pour les données de volume horaire et la numérotation automatique.
Matplotlib 3.9.2 ★ 18.1k 	Bibliothèque de visualisation de données en Python qui produit des graphiques de qualité publication	gère la création des visualisations des AAT à travers des graphiques circulaires représentant les compétences, avec une palette de couleurs personnalisée. La bibliothèque assure l'export en format vectoriel pour une intégration de haute qualité dans les PDFs, tout en permettant une personnalisation fine des styles graphiques.

Make 4.4.1 ★5.1k (GNU Make) 	Outil d'automatisation de la compilation et de l'exécution de tâches	<p>Orchestration du processus de build complet :</p> <ul style="list-style-type: none"> génération des fichiers JSON, conversion Markdown vers Typst, compilation des PDF. Gestion des dépendances entre les différentes étapes de génération.
Vitepress ★ 9.9k 	Framework de génération de sites statiques basé sur Vue.js, conçu spécialement pour la documentation technique	VitePress génère le site statique multilingue à partir des fichiers Markdown structurés. Il intègre les composants Vue.js personnalisés comme ChartDonut pour les AAT et les tableaux dynamiques. La gestion des données est assurée par gray-matter pour les métadonnées YAML et chart.js pour les graphiques, offrant une expérience utilisateur moderne et responsive.
Typst 0.11.1 ★ 23.4k 	Système moderne de composition de documents, conçu comme une alternative moderne à LaTeX	Prend en charge la génération des PDF des programmes pédagogiques avec une mise en page professionnelle et une gestion avancée des styles. Il utilise des templates personnalisés pour les différentes sections du document, avec une gestion cohérente des styles et de la typographie. L'intégration dynamique permet d'incorporer les données JSON dans les templates et de générer automatiquement les éléments structurels comme les tables des matières.

Table 1: Outils et Technologies du Projet

3.1 Analyse comparative des alternatives

Python 3.12.6

Critère	Python	Java	JavaScript	C++
Facilité d'utilisation	Excellente	Complexe	Bonne	Très complexe
Temps de développement	Rapide	Long	Moyen	Long
Support intégré web	Excellent	Complexe	Natif	Limité
Classement TIOBE	1	2	7	4

Table 2: Python et ses alternatives

Le choix de ces quatre langages pour la comparaison est justifié par leur position dominante dans l'index TIOBE de janvier 2024, où Python occupe la première place avec 15.16% de part de marché, suivi de Java (11.72%), C++ (9.96%) et JavaScript (3.17% - 7ème place). Bien que d'autres langages comme C(5) ou PHP (6) soient également populaires, ils n'ont pas été retenus car soit trop bas niveau (C), soit trop liés à un écosystème spécifique (C avec Microsoft), soit trop spécialisés (PHP pour le web). Python s'impose comme le choix optimal pour notre projet grâce à sa position dominante actuelle, sa facilité d'utilisation, sa rapidité de développement, son excellent support des technologies web et son écosystème scientifique robuste, offrant ainsi le meilleur compromis entre performance, accessibilité et polyvalence par rapport aux alternatives disponibles.

Miniconda

Critère	Miniconda	Anaconda	venv/virtualenv	Poetry
Taille installation	400MB	>3GB	< 50MB	100MB
Gestion dépendances	Complète	Complète	Basique	Avancée
Multi-plate-forme	Totale	Totale	Limitée	Bonne

Table 3: Miniconda et ses alternatives

Dans notre choix d'outil de gestion d'environnement Python, Miniconda s'impose comme la solution optimale, offrant un équilibre parfait entre légèreté et fonctionnalités complètes.

Contrairement à Anaconda avec ses gigaoctets de packages préinstallés souvent superflus, ou aux alternatives comme venv et virtualenv qui se limitent aux packages Python purs, Miniconda nous permet de gérer efficacement à la fois les packages Python et les dépendances binaires essentielles comme Pandoc et Typst. Bien que Poetry propose une approche moderne, sa spécialisation dans le développement de packages Python le rend moins adapté à nos besoins, tandis que la maturité et la stabilité de l'écosystème Conda garantissent une fiabilité à long terme pour notre projet.

Flask

Critère	Flask	Django	FastAPI	Pyramid
Taille installation	Légère(600KB)	Lourde(>10MB)	Légère(2MB)	Légère(400KB)
Gestion dépendances	Complète	Complète	Basique	Avancée
Courbe d'apprentissage	Simple	Complexe	Modérée	Modéré

Table 4: Flask et ses alternatives

Dans notre choix de framework web, Flask s'est distingué comme la solution idéale grâce à sa légèreté et sa flexibilité exceptionnelle. Contrairement à Django qui impose une structure rigide avec de nombreuses fonctionnalités superflues, ou à FastAPI qui est principalement orienté API REST, Flask nous offre exactement ce dont nous avons besoin pour notre interface web, avec une intégration native de Jinja2 et une courbe d'apprentissage douce. Pyramid, bien que complet, propose une approche plus complexe qui dépasserait nos besoins actuels pour une simple interface de configuration et de prévisualisation.

Jinja2

Critère	Jinja2	Mako	Django Templates	Mustache
Performance	Excellente	Bonne	Django-only	Limitée
Intégration Python	Native	Complète	Basique	Avancée
Documentation	Très complète	Bonne	Très bonne	Basique

Table 5: Jinja2 et ses alternatives

Le choix de Jinja2 comme moteur de template s'est naturellement imposé par sa puissance et son intégration parfaite avec notre écosystème Python. Comparé à d'autres moteurs de template comme Mako ou Cheetah qui offrent des fonctionnalités plus limitées, ou à des solutions plus complexes comme Django Templates qui sont étroitement liées à leur framework, Jinja2 offre le meilleur compromis entre flexibilité et simplicité, tout en maintenant

des performances excellentes et une syntaxe claire pour la création de nos templates de documents.

Vitepress

Critère	Vitepress	Docusaurus	MkDocs	Sphinx
Temps de build	< 1s	2-3s	1-2s	>3s
Framework	Vue.js	React	Moyenne	Très complexe
Configuration	Simple	complexe	Moyenne	Très complexe

Table 6: Vitepress et ses alternatives

Pour la génération de notre site de documentation, VitePress représente la solution la plus adaptée, combinant performance exceptionnelle et simplicité d'utilisation. Contrairement à Docusaurus qui nécessite une configuration complexe avec React, ou à MkDocs qui manque de fonctionnalités modernes, VitePress offre un excellent support du Hot Module Replacement et une intégration native avec Vue.js. Sphinx, bien que puissant, aurait introduit une complexité excessive pour nos besoins de documentation technique.

Make

Critère	Make	Gradle	npm scripts	Ant
Dépendances	Minimales	Java	Node.js	Java
Taille	Légère (1MB)	Lourde (>100MB)	Moyenne	Moyenne
Vitesse exécution	Très rapide	Moyenne	Moyenne	Lente

Table 7: Make et ses alternatives

Make s'est révélé être l'outil d'automatisation idéal pour notre projet, offrant une solution éprouvée et universelle pour la gestion des builds. Comparé à des alternatives comme Gradle qui nécessite Java et une configuration complexe, ou npm scripts qui dépend de Node.js, Make propose une approche simple et efficace sans dépendances supplémentaires. Sa disponibilité universelle et sa capacité à gérer efficacement les dépendances entre les différentes étapes de notre processus de génération en font l'outil parfait pour notre workflow.

Pandoc

Critère	Pandoc	python-markdown	CommonMark	Mistune
Formats supportés	>40 formats	Markdown uniquement	Markdown strict	Markdown uniquement
Performance	Bonne	Très bonne	Excellente	Excellente
Extensibilité	Excellente (Lua)	Moyenne (Python)	Limitée	Moyenne

Table 8: Pandoc et ses alternatives

Pandoc s'impose comme le choix optimal pour notre projet grâce à sa polyvalence et son extensibilité incomparable via les filtres Lua, qui gère la transformation complexe des métadonnées et le formatage structuré (entêtes, sections, tableaux), Pandoc peut être précisément adapté à nos besoins grâce à son système de filtres. Contrairement à python-markdown ou Mistune qui se limitent au format Markdown, Pandoc gère plus de 40 formats différents et permet une personnalisation poussée de la conversion, notamment pour les tables, les figures et la gestion de la mise en page. Bien que CommonMark soit plus rapide pour le Markdown pur, sa limitation à une implémentation stricte ne convient pas à nos besoins de transformation complexes qui nécessitent la flexibilité offerte par les filtres Lua de Pandoc.

Typst

Critère	Typst	Word	AsciiDoc	LateX
Temps compilation	< 1s	N/A	2s	>10s
Programmabilité	Native	Limitée	Bonne	Extensive
Installation	100MB	>1GB	50MB	>2GB

Table 9: Typst et ses alternatives

Typst se démarque par sa rapidité de compilation exceptionnelle et sa programmabilité native, essentielles pour notre génération automatisée de documents. Contrairement à Word qui manque de capacités d'automatisation robustes, ou à LaTeX qui ne garantit pas une qualité typographique constante pour les PDF, Typst offre le meilleur compromis entre performance et qualité. AsciiDoc, bien que puissant, n'atteint pas le niveau de programmabilité et d'intégration que nous offre Typst.

Matplotlib

Critère	Matplotlib	Plotly	Bokeh	Seaborn
Export PDF	Excellent	Limité	Moyen	Hérite Matplotlib
Documentation	Exhaustive	Très bonne	Bonne	Très bonne
Performance	Très bonne	Bonne	Bonne	Très bonne

Matplotlib reste incontournable pour notre projet grâce à son contrôle total sur le rendu et son excellente qualité d'export PDF, essentiels pour la génération de nos graphiques circulaires représentant les AAT. Plotly et Bokeh, bien qu'excellents pour les visualisations web interactives, ne conviennent pas à nos besoins d'export PDF statique de haute qualité. Seaborn, construit sur Matplotlib, ajoute une couche d'abstraction inutile pour nos besoins spécifiques de visualisation des AAT qui nécessitent un contrôle précis sur les aspects graphiques comme les couleurs, la disposition et l'export vectoriel.

Conclusion

L'architecture technique développée pour ce projet repose sur une sélection rigoureuse d'outils open source modernes et performants. Python constitue le cœur du système, orchestrant l'ensemble du processus de génération de documents, tandis que Miniconda assure la reproductibilité de l'environnement de développement. La chaîne de traitement documentaire, constituée de Pandoc, Jinja2 et Typst, offre une solution moderne et puissante pour la génération de documents de haute qualité, complétée par Flask pour l'interface web et VitePress pour la documentation. Cette architecture démontre qu'il est possible de construire un système robuste en s'appuyant sur des technologies modernes, tout en maintenant une complexité maîtrisée. Les choix technologiques effectués répondent aux besoins actuels de l'ENIB en matière de génération de documents pédagogiques, tout en posant les bases d'évolutions futures grâce à leur flexibilité et leur interopérabilité. Le chapitre 4 sera consacré à l'étude approfondie de Typst, en examinant ses mécanismes internes et sa syntaxe innovante pour la composition de documents académiques.

CHAPITRE 4 : Implémentation de Typst

Introduction

La rédaction de documents techniques et académiques nécessite des outils adaptés aux exigences modernes. Dans ce contexte, Typst émerge comme une solution novatrice, proposant une approche repensée de la composition de documents. Ce chapitre présente une analyse approfondie de cet outil, depuis ses origines jusqu'à son application concrète dans notre contexte académique. Notre étude s'articule en trois axes majeurs : une exploration des fondements et fonctionnalités de Typst, une analyse comparative avec LaTeX mettant en lumière ses avantages distinctifs, et enfin la présentation de notre solution de modèle pour les rapports de stage de l'ENIB.

4 Comprendre Typst : Origines et Fonctionnalités

4.1 La Naissance de Typst

La création de Typst en 2019 par Martin Haug et Laurent Kirsch marque un tournant dans l'histoire des systèmes de composition de documents [17]. Cette initiative est née d'une observation critique du paysage des outils de rédaction technique, dominé depuis les années 1980 par LaTeX [18]. Comme l'expliquent Wilson et al.[19], l'évolution des besoins en matière de rédaction technique et académique nécessitait une approche plus moderne de la composition de documents. Plusieurs facteurs ont motivé ce développement :

4.1.1 Contexte Historique

- 1970s-1980s : Domination de TeX et LaTeX
- 1990s-2000s : Tentatives d'amélioration avec LyX et autres éditeurs WYSIWYG
- 2010s : Émergence des besoins modernes (collaboration en temps réel, prévisualisation instantanée)
- 2019 : Lancement de Typst comme réponse à ces nouveaux besoins

4.1.2 Evolution du Projet

Le développement de Typst s'est structuré autour de plusieurs phases clés :

1. Phase Alpha (2019-2020) : Développement du moteur de rendu
2. Phase Bêta (2020-2021) : Implémentation des fonctionnalités de base
3. Version 1.0 (2023) : Première version stable avec support complet des fonctionnalités essentielles
4. Développement Continu : Ajouts réguliers de nouvelles fonctionnalités et optimisations

4.2 Contexte et Point de Départ

Système LaTeX Initial

Le système de génération de PDF de l'ENIB reposait initialement sur une architecture développée par Monsieur Vincent Choqueuse. Son approche se basait sur LaTeX pour produire les PDF de la formation FISEA. La chaîne de traitement commençait avec des fichiers Markdown enrichis d'en-têtes YAML contenant les métadonnées essentielles des cours : volumes horaires, coefficients, rattachements aux UE. Un script Python traitait ces fichiers Markdown pour générer des fichiers TeX individuels pour chaque cours et chaque semestre.

Le fichier central main.tex, également généré par le script Python, orchestrerait l'assemblage de tous ces composants. Il importait les packages LaTeX requis, définissait la mise en page globale du document, et incluait systématiquement tous les fichiers TeX des cours et des semestres dans le bon ordre. Chaque fichier TeX de cours contenait la structure complète d'un enseignement, avec ses tableaux de volumes horaires, sa description et ses acquis d'apprentissage. Les fichiers TeX des semestres, quant à eux, présentaient les synthèses semestrielles avec les totaux (ECTS, volumes horaires, etc.).

Cette approche modulaire permettait une génération automatisée des documents finaux, mais la syntaxe complexe de LaTeX, avec ses nombreuses commandes basées sur des backslashes, rendait le système difficile à maintenir et à faire évoluer. Les fichiers TeX générés étaient volumineux, peu attrayants, et nécessitaient beaucoup de temps pour la génération.

4.3 Première Phase : Adaptation Initiale vers Typst

Dans un premier temps, la structure fondamentale du projet a été maintenue. Les fichiers Markdown sont restés la source primaire des données, et Pandoc a conservé son rôle central dans l'analyse de ces fichiers. Les templates Jinja2 ont également été adaptés, et les scripts Python existants ont été modifiés (create_main.py) ou complétés par un nouveau script (create_courses.py). L'arborescence du projet conservait sa logique de séparation entre les dossiers.



Figure 6: Commande tree pour le premier projet (LaTeX)

```

build
├── courses_fise
├── courses_fisea
├── home
└── images
    └── semester_fise
        └── semester_fisea
config
data
└── content
    ├── S10
    ├── S1A
    ├── S2P
    ├── S3A
    ├── S4A
    ├── S4P
    ├── S5A_S5P
    ├── S50
    ├── S5_FISEA
    ├── S6A
    ├── S60
    ├── S6P
    ├── S6_FISEA
    ├── S7A_S7P
    ├── S7_FISEA
    ├── S8
    ├── S8_FISEA
    ├── S9A_S9P
    └── S9_FISEA
pandoc
templates
tools

```

Figure 7: Commande tree pour le nouveau projet (Typst)

4.4 Adaptation des Templates et du Système de Filtrage

La transposition des fichiers de mise en forme LaTeX vers Typst a marqué une étape clé dans la modernisation du système de documentation. L'ancien système reposait sur trois fichiers principaux: main.tex, semester.tex et apc.tex, chacun jouant un rôle spécifique dans l'organisation et la présentation des documents PDF. Main.tex assurait la structure globale du document, en définissant les paramètres de mise en page, la table des matières, la couverture, et en orchestrant l'inclusion des fichiers des cours et semestres. Semester.tex se concentrerait sur la mise en forme des informations propres à chaque semestre, notamment les totaux des volumes horaires, des crédits ECTS et les synthèses associées. Apc.tex était dédié à l'Approche par Compétences, structurant les tableaux des acquis d'apprentissage et des compétences liées aux cours.

Ces fichiers ont été remplacés par main.typ, semester.typ et tables.typ, chacun conçu pour exploiter les capacités modernes de Typst tout en simplifiant leur maintenance. Main.typ conserve son rôle central en définissant la structure principale du document et en orchestrant l'inclusion des fichiers Typst pour les cours et semestres. Semester.typ reprend la gestion des informations semestrielles en les adaptant à une syntaxe plus lisible et maintenable. Tables.-typ a été introduit pour gérer spécifiquement les tableaux, notamment ceux des

volumes horaires et des acquis d'apprentissage, en exploitant la flexibilité de Typst pour des présentations plus claires et homogènes.

La syntaxe de Typst a permis de simplifier considérablement le système. Les sections et sous-sections, autrefois définies par des commandes complexes comme `section` et `subsection` en LaTeX, utilisent désormais une notation intuitive avec le symbole `#`. Les tableaux, auparavant gérés par des environnements lourds tels que `begin{table}` et `end{table}`, ont été remplacés par des structures plus légères grâce à la syntaxe native de Typst. Ces nouvelles structures permettent de définir directement les colonnes, les en-têtes, et les lignes avec des listes ou des blocs structurés, sans nécessiter de nombreuses commandes intermédiaires.

Dans LaTeX, créer un tableau nécessitait un environnement dédié et plusieurs commandes pour délimiter les colonnes, insérer des lignes et définir des styles. Voici un exemple de tableau en LaTeX:

```

2   \centering
3   \begin{tabular}{|c|c|c|c|}
4     \hline
5     UE & Type & Volume \\ \hline
6     S5UE1 & CM & 20h \\ \hline
7     S5UE2 & TD & 10h \\ \hline
8   \end{tabular}
9   \caption{Exemple de tableau en LaTeX}

```

Figure 8: Exemple de tableau en LaTeX

Ce tableau nécessite plusieurs commandes, comme `hline` pour insérer des lignes horizontales, et une définition rigide des colonnes (`{|c|c|c|}`) qui limite la flexibilité. De plus, tout ajustement, tel que la largeur des colonnes, demande des modifications supplémentaires dans la structure du tableau, rendant son maintien et sa personnalisation complexes.

En revanche, Typst propose une approche bien plus intuitive et concise pour créer des tableaux. Avec Typst, voici comment le même tableau serait écrit:

```

1 #table(
2   columns: [auto, 4cm, auto],
3   rows: [
4     ["UE", "Type", "Volume"],
5     ["S5UE1", "CM", "20h"],
6     ["S5UE2", "TD", "10h"]
7   ]
8 )

```

Figure 9: Exemple de tableau en Typst

La syntaxe Typst permet de définir les colonnes (par exemple, avec des tailles fixes comme 4cm ou automatiques avec auto) et d'ajouter les lignes directement dans une liste structurée. Les en-têtes, données, et mises en page sont séparés de manière claire, ce qui améliore considérablement la lisibilité et la maintenabilité du code.

Le script filter.lua, utilisé dans le processus de transformation via Pandoc, a également été révisé en profondeur. Sa fonction principale consiste à convertir les éléments Markdown et les métadonnées associées en syntaxe Typst. Là où l'ancien filtre produisait des commandes LaTeX comme `includegraphics` ou `begin{table}`, la nouvelle version génère des équivalents Typst concis et modernes. Par exemple, les images sont désormais intégrées via la commande `#figure`, tandis que les tableaux bénéficient d'une syntaxe structurée simplifiée.

Cette transition a permis non seulement de réduire la complexité du système, mais aussi d'augmenter sa lisibilité et sa maintenabilité, tout en adoptant une approche plus flexible et adaptée aux besoins actuels.

4.5 Structuration des Données et Organisation du Projet

La réorganisation de l'architecture du projet a constitué une étape majeure de la migration. Le dossier build initial regroupait l'ensemble des fichiers générés dans des sous-dossiers comme courses et semester. Cette organisation, adaptée à une seule formation, a dû évoluer pour accueillir la formation FISE. De nouveaux dossiers ont été créés : courses_fise et courses_fisea pour les cours, semester_fise et semester_fisea pour les semestres, permettant une séparation claire des contenus par formation, les graphiques générés par Matplotlib sont stockées aussi dans les dossiers de cours correspondants.

4.6 Centralisation des Données et Automatisation

Un nouveau script Python a été développé pour centraliser l'ensemble des informations du système. Ce script analyse systématiquement les en-têtes YAML des fichiers Markdown ainsi que les fichiers JSON auxiliaires (`aat.json` et `course.json`) qui décrivent les acquis

d'apprentissage et la structure des cours. Il génère ensuite deux fichiers JSON principaux, fise.json et fisea.json, qui servent de référence unique pour toutes les informations nécessaires à la génération des documents.

Cette approche de "source unique de vérité" garantit une cohérence parfaite entre les différentes utilisations des données. Toute modification dans un fichier source se répercute automatiquement lors de la compilation suivante, assurant ainsi la synchronisation des informations à travers tout le système.

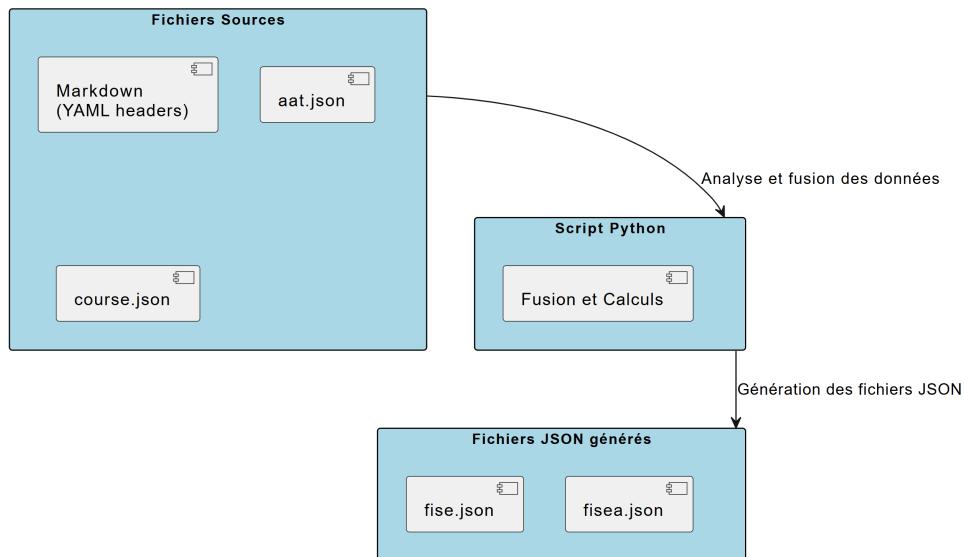


Figure 10: Génération Centralisée des Fichiers JSON(FISE/FISEA)

4.7 Optimisation et Contrôle de la Génération

Un fichier de configuration YAML a été implémenté pour offrir un contrôle fin sur la génération des documents. Il permet notamment de gérer l'inclusion ou l'exclusion de sections globales comme l'Approche par Compétences, d'activer ou de désactiver l'affichage de certains éléments comme les tableaux de volumes horaires, et de personnaliser la présentation des schémas dans les pages de cours.

Un Makefile complet a été développé pour simplifier l'utilisation du système. Il offre différentes cibles comme make json pour la génération des fichiers JSON, make typst pour la création des fichiers Typst, et make pdf pour la compilation finale. Cette approche permet aux utilisateurs non techniques de générer facilement les documents sans avoir à maîtriser les détails techniques du système.

```
frontpage:
  title: PROGRAMME DE FORMATION
  subtitle: Nom Prénom
  academic_year: 2024/2025
  contact:
    name: Hello
    email: entreprises@enib.com
    phone: 0298033878
  toc:
    depth: 2
  display_settings:
    show_semester: true
    show_selected_courses: true
    semester:
      default:
        show_information_table: true
        show_content_table: true
      courses:
        default:
          show_general_info: true
          show_volume_table: true
          show_aat_chart: true
          general_info_columns:
            ue: true
            type: true
            coeff: true
            volume_estimated: true
            volume_edt: true
    content:
      add_introduction: false
      include_sections:
        organisation: false
        apc: false
        intersemestres: false
        sessions: false
        bac2: false
        annees4_5: false
      semesters:
        include:
        - '*'
        exclude:
      courses:
        include:
        exclude: []
```

Figure 11: fichier de configuration YAML

4.8 Performances et Résultats

Les tests de performances réalisés sur le document TemplateContratsPro_2024-2025 démontrent des améliorations significatives. Le temps de compilation est passé de 4,469 secondes avec LaTeX à 0,852 seconde avec Typst, soit une réduction de plus de 80%. L'empreinte système a été considérablement réduite, passant de 4-7 Go pour une installation LaTeX complète à seulement 20-30 Mo pour Typst. La taille des documents PDF générés a légèrement augmenté, passant de 0,470 Mo à 0,623 Mo, une augmentation de 32%, principalement due à l'incorporation complète des polices, mais qui reste acceptable au regard des gains en performances et en facilité d'utilisation.

4.9 Illustration du Processus de Génération des Documents PDF avec Typst

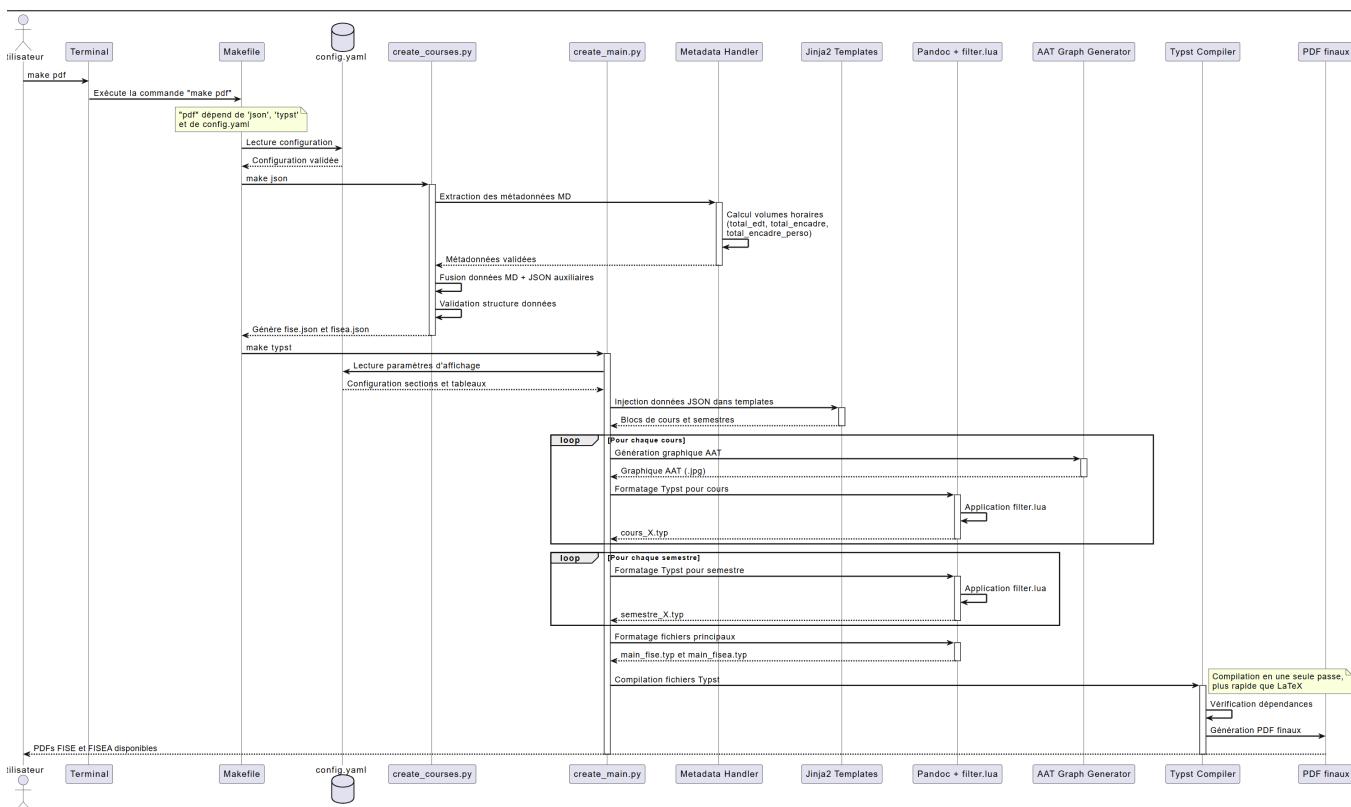


Figure 12: Diagramme de séquence- génération de documents PDF avec Typst

La Figure 12 illustre le flux général de génération des documents pédagogiques PDF utilisant Typst. L'utilisateur déclenche l'ensemble du processus en exécutant la commande make pdf. Cette commande active successivement les cibles du Makefile, qui coordonne

l'analyse des métadonnées, la génération des fichiers JSON centralisés (fise.json et fisea.json), et la production des fichiers Typst pour chaque cours et semestre. Ces étapes conduisent à la compilation finale par Typst, produisant les documents PDF finaux.

4.10 Développement d'un Template de Rapport de Stage

Suite à la migration réussie vers Typst, nous avons développé un template moderne et flexible pour la rédaction des rapports de stage des étudiants de l'ENIB. Ce template exploite pleinement les capacités de Typst tout en maintenant les standards académiques de l'école.

4.10.1 Architecture du Template

Le template s'organise autour d'une structure modulaire permettant une séparation claire des responsabilités :

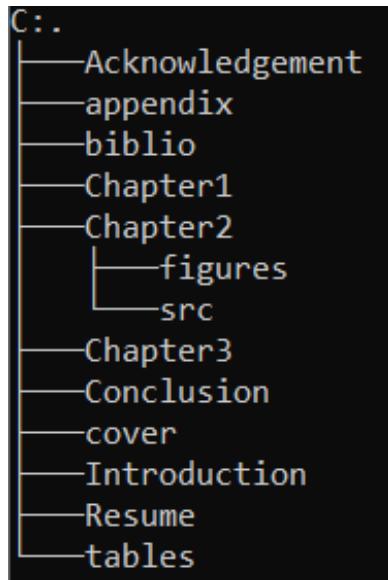


Figure 13: Commande tree pour le template de rapport

4.10.2 Le Fichier Principal (main.typ)

Le fichier main.typ constitue le point d'entrée du template et orchestre l'ensemble du document. Il commence par importer le module de la page de couverture et définit les paramètres fondamentaux de mise en page. Une caractéristique notable est la gestion de la confidentialité via une variable booléenne qui, lorsqu'elle est activée, applique un filigrane "CONFIDENTIEL" en rotation sur chaque page. La configuration de base utilise la police Segoe UI en taille 12pt et définit des marges spécifiques pour le document.

La structure du document est ensuite organisée de manière méthodique avec une séparation claire entre les différentes parties. Le fichier gère intelligemment la numérotation des sections en la désactivant pour les parties préliminaires (remerciements, tables), l'activant pour le corps principal du rapport (introduction et chapitres), puis la désactivant à nouveau pour les parties finales (bibliographie, annexes, résumé). Cette approche modulaire permet d'inclure séparément chaque partie du rapport via des fichiers dédiés, facilitant ainsi la maintenance et la clarté du code.

4.11 Conclusion

La migration du système de génération de PDF de l'ENIB, passant de LaTeX à Typst, a marqué une étape clé dans la modernisation de la gestion documentaire. Cette transition a permis d'améliorer significativement la lisibilité, la maintenabilité et les performances du processus tout en répondant aux besoins croissants de l'institution.

Grâce à Typst, le système bénéficie d'une syntaxe intuitive qui remplace les commandes complexes de LaTeX, rendant la génération des documents plus rapide et plus accessible. Les nouveaux templates adaptés aux formations FISE et FISEA offrent une gestion claire des sections, tableaux et graphiques, tout en minimisant les risques d'erreurs.

La centralisation des données via les fichiers JSON (fise.json et fisea.json), générés automatiquement par des scripts Python, assure une parfaite cohérence entre les différentes utilisations des informations pédagogiques. Par ailleurs, l'intégration d'un fichier de configuration YAML et d'un Makefile complet simplifie grandement l'utilisation pour des utilisateurs non techniques.

Les tests de performances ont confirmé des gains significatifs: une réduction de plus de 80% du temps de compilation et une empreinte système bien plus légère, passant de plusieurs gigaoctets à quelques dizaines de mégaoctets. Ces améliorations s'accompagnent d'une flexibilité accrue pour les futures évolutions du système.

En somme, cette migration ne constitue pas seulement une adaptation technique, mais représente une véritable transformation, dotant l'ENIB d'un système documentaire moderne, robuste et évolutif, parfaitement adapté aux exigences pédagogiques et administratives actuelles.

CHAPITRE 5 : Développement de l'interface utilisateur

Introduction

Après avoir migré avec succès la génération des documents PDF vers Typst et mis en place un système de configuration flexible, l'étape suivante consistait à développer des interfaces utilisateur adaptées aux besoins du personnel administratif de l'ENIB. La manipulation directe des fichiers de configuration YAML pour la personnalisation des PDF et la gestion des métadonnées des fichiers Markdown pouvait s'avérer source d'erreurs et ralentir le processus de mise à jour du programme pédagogique.

Ce chapitre présente les différentes interfaces développées pour faciliter la gestion du programme pédagogique. Nous aborderons d'abord la création d'un outil en ligne de commande pour la gestion des métadonnées YAML, puis le développement d'une interface web intuitive avec Flask pour la personnalisation des documents PDF, et enfin l'intégration de la formation FISEA dans le site statique VitePress existant.

5 Gestion des métadonnées YAML

Les fichiers Markdown qui composent le programme pédagogique comportent, en en-tête, des métadonnées YAML indispensables : acquis d'apprentissage (AAT), coefficients, volumes horaires, tags, etc. Pour éviter les erreurs de saisie et fluidifier la maintenance, un outil interactif a été développé en Python. Il s'appuie notamment sur PyYAML pour analyser et modifier le YAML, colorama pour afficher en couleur les informations et avertissements (vert pour les succès, rouge pour les erreurs, jaune pour les avertissements), et dataclasses pour structurer plus clairement le code.

5.1 Structure et organisation du code

Pour clarifier la structure et renforcer la maintenabilité, le projet est réparti en trois fichiers principaux. Le premier, `models.py`, définit les classes de données. La classe `CourseMetadata` regroupe toutes les informations d'un cours (identifiant, coefficient, volumes, tags, etc.) et autorise l'ajout, la modification ou la suppression de champs, tout en gardant une trace des changements. La classe `Volume` se consacre à la gestion des volumes horaires, comme le CM, le TD, le TP ou le travail personnel, en assurant la cohérence et les conversions nécessaires (CM, TD, etc. en texte, `total_edt` en nombre décimal).

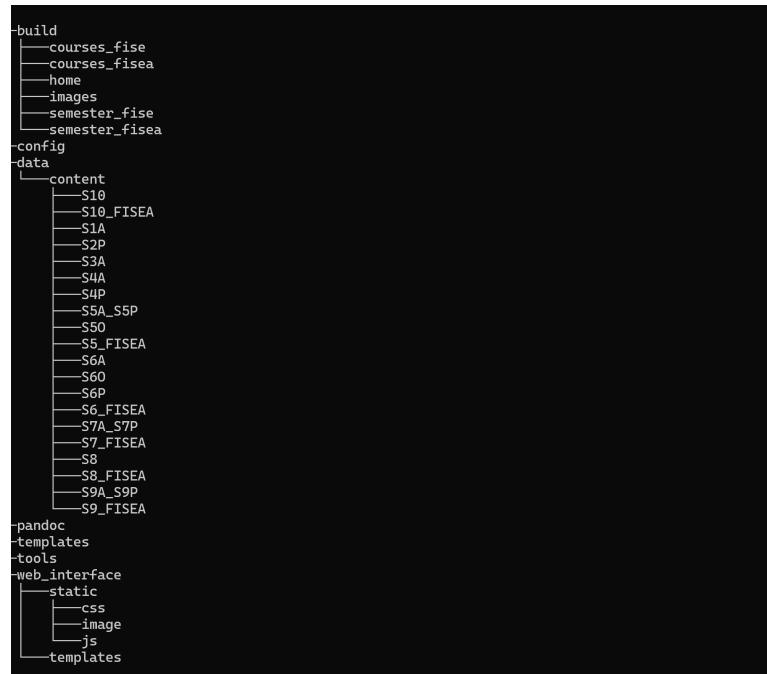


Figure 14: Résultat de la Commande 'tree' : Structure des Répertoires

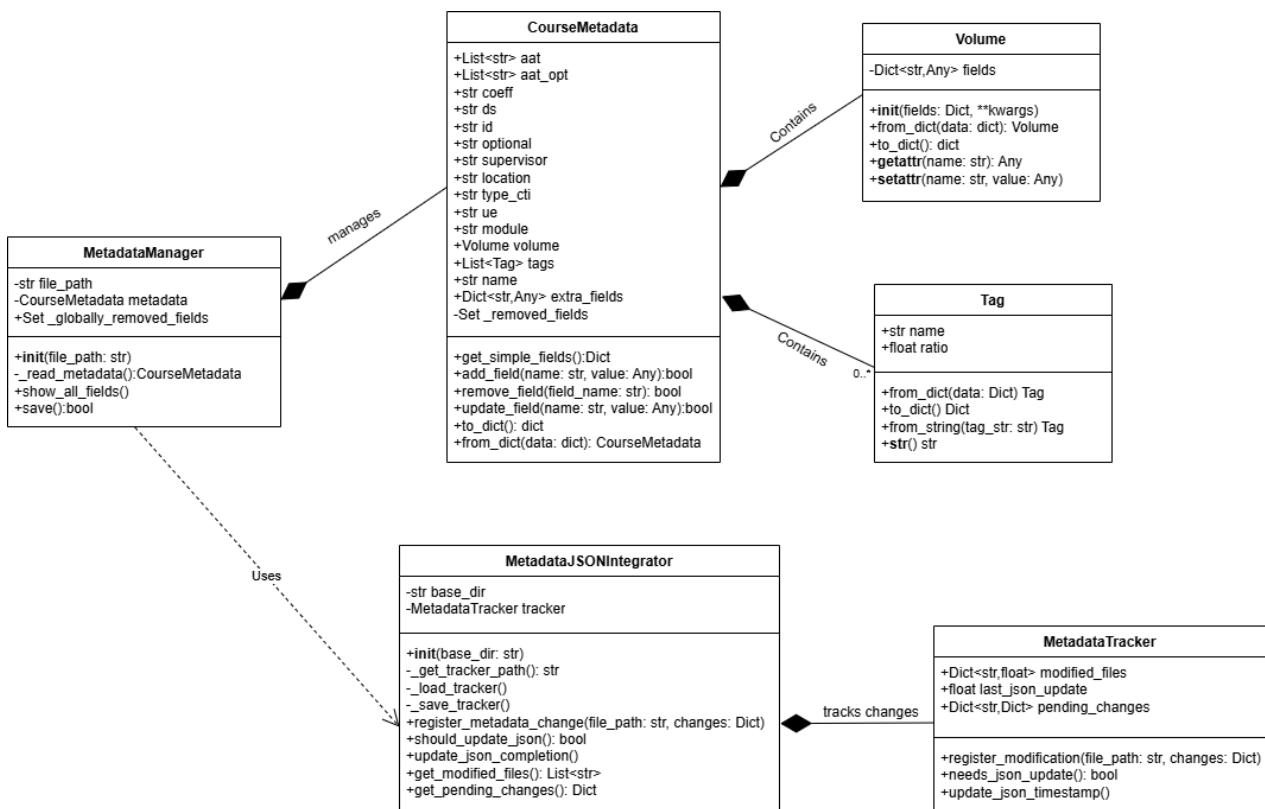


Figure 15: Diagramme de classes du Système de Gestion des Métadonnées YAML

Le second fichier, managers.py contient la logique de lecture et d'écriture des fichiers Markdown, assurée par la classe MetadataManager. Celle-ci fait le lien avec CourseMetadata et Volume, veille à la cohérence du format de sortie et gère la détection et la résolution d'erreurs (fichier introuvable, syntaxe YAML incorrecte, etc.). Un mécanisme de type singleton est également mis en place pour l'intégration JSON : une fonction get_integrator() crée une unique instance de MetadataJSONIntegrator, puis la réutilise systématiquement. Ce fonctionnement évite la création de multiples instances concurrentes et prévient les rechargements ou opérations coûteuses répétées, tout en réduisant le risque de surcharge système.

Le troisième fichier, cli.py, regroupe la logique d'interface en ligne de commande : les menus, les saisies et l'affichage coloré des messages. C'est dans ce fichier que l'utilisateur navigue entre les différents modes (modification d'un seul fichier ou opérations en lot), tandis que les retours visuels (succès, avertissements, erreurs) sont gérés.

Cette organisation présente plusieurs avantages. Les classes de données (dans models.py) sont clairement séparées de la logique métier (dans managers.py) et de l'interface CLI (dans cli.py). Chacune de ces parties peut évoluer de manière indépendante. Il est par exemple envisageable de changer complètement l'interface (passer d'une CLI à une application Flask) sans modifier la gestion interne des métadonnées.

5.1.1 Choix de la formation et modes d'édition

L'application invite d'abord l'utilisateur à choisir la formation qu'il souhaite modifier (FISE ou FISEA), comme l'illustre la Figure 16. Une fois ce premier choix effectué, elle propose deux modes : la modification d'un seul fichier à la fois ou la réalisation d'opérations en lot sur l'ensemble des cours de la formation sélectionnée. Dans le premier cas, l'utilisateur opte pour un fichier précis et accède à plusieurs menus (champs simples, volumes, tags) afin d'ajouter, de modifier ou de supprimer des informations de manière structurée. Dans le second cas, il peut réaliser des opérations groupées, par exemple l'ajout d'un champ à tous les fichiers ou la suppression d'un champ obsolète.

```
C:\Users\ayoub\Desktop\please\md_to_tpyst\tools>python manage_metadata.py

Select mode:
1. FISE
2. FISEA

Choice (1-2): 2

Select operation:
1. Modify individual file
2. Modify all files in batch
3. Return to mode selection

Choice (1-3): |
```

Figure 16: Interface de sélection du mode de modification.

5.1.2 Modification individuelle

Lorsque l'utilisateur choisit la modification d'un seul fichier, une liste numérotée des fichiers disponibles est présentée. Il sélectionne alors un cours précis et accède à un menu structuré en trois sections : la gestion des champs standards (identifiant, coefficient, type), la modification des volumes horaires (CM, TD, TP, projet) et la manipulation des tags. Chaque section autorise la visualisation des valeurs actuelles, l'ajout de nouveaux éléments, la modification et la suppression.

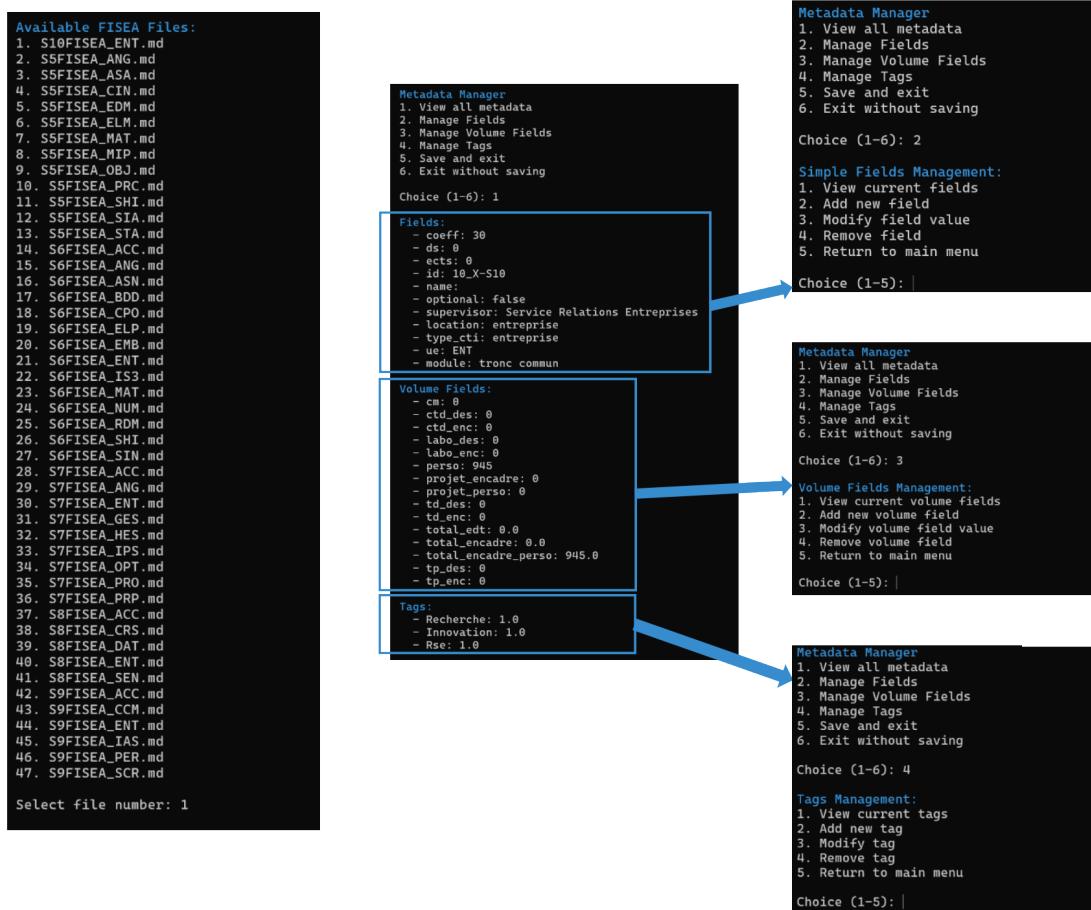


Figure 17: Structure et navigation de l'outil de gestion des métadonnées

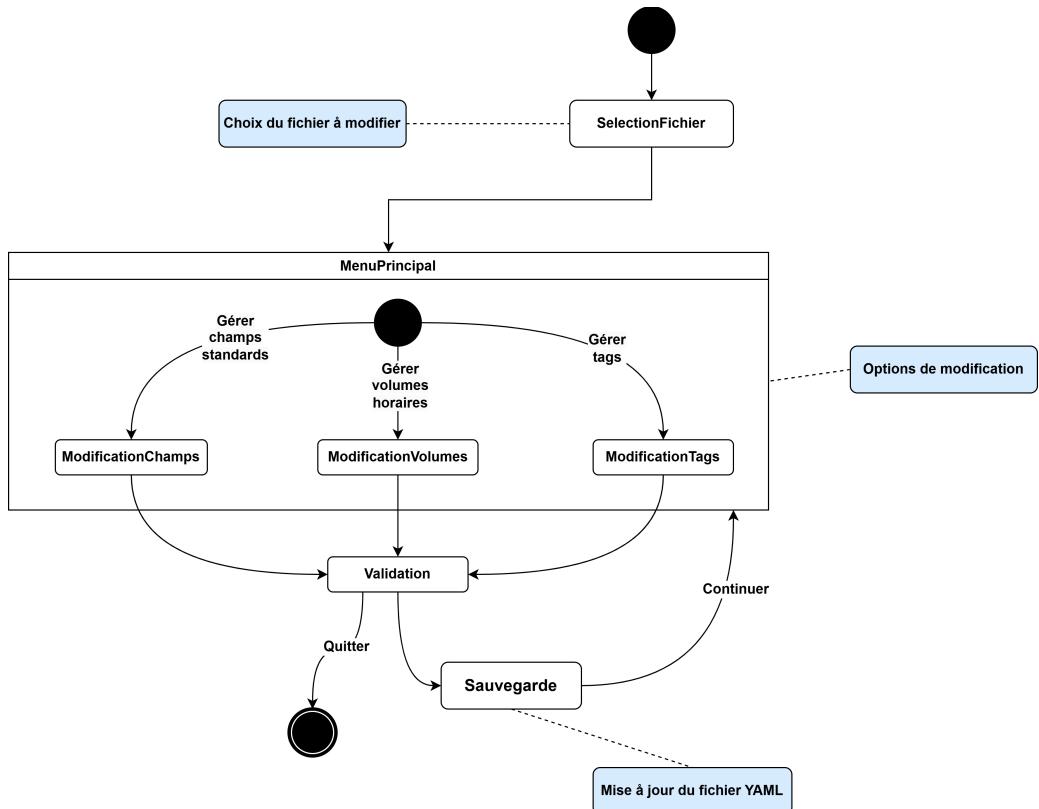


Figure 18: Diagramme d'État – Mode Individuel de Modification des Fichiers YAML

5.1.3 Opérations en lot

Les opérations en lot autorisent la manipulation simultanée de plusieurs fichiers. Sept types d'opérations sont proposés : ajouter ou modifier des champs standards, supprimer des champs, gérer les volumes horaires (ajout, modification, suppression), ajouter des tags ou en supprimer. Avant chaque opération, l'outil informe du nombre de fichiers concernés et demande une confirmation. L'utilisateur peut également visualiser un exemple représentatif des métadonnées afin de mieux anticiper l'impact de sa modification.

```

Select mode:
1. FISE
2. FISEA

Choice (1-2): 2

Select operation:
1. Modify individual file
2. Modify all files in batch
3. Return to mode selection

Choice (1-3): 2

Found 47 FISEA files
Proceed with batch operations? (y/n): y

Batch Operations Menu (FISEA - 47 files):
1. Add/Modify field in all files
2. Remove field from all files
3. Add/Modify volume field
4. Remove volume field
5. Add tag to all files
6. Remove tag from all files
7. View sample metadata
8. Return to main menu

Choice (1-8): 1
  
```

Figure 19: Menu des opérations par lot

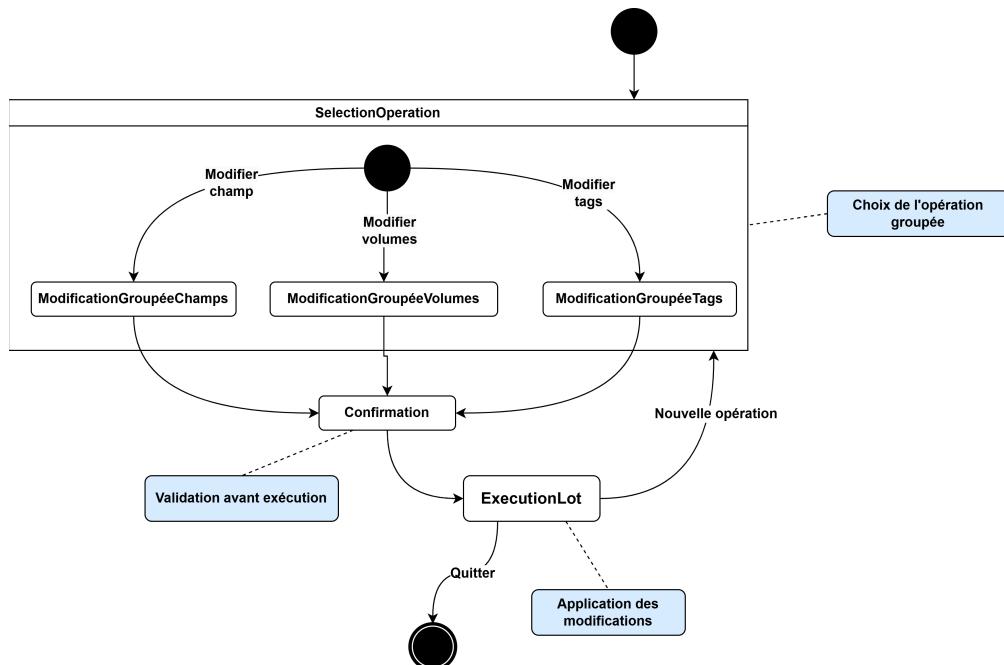


Figure 20: Diagramme d'État – Processus de Modification en Mode Groupé

Cette approche renforce à la fois la cohérence et la rapidité de mise à jour, les utilisateurs peuvent effectuer des modifications massives en quelques clics. De plus, la séparation du code en trois fichiers (models, managers, cli) facilite la maintenance et permet d'envisager facilement l'évolution de l'outil, qu'il s'agisse d'intégrer de nouveaux champs, de remplacer la CLI par une interface web ou d'étendre les fonctions de contrôle d'erreurs.

5.2 Interface Web avec Flask

La manipulation directe du fichier de configuration YAML pour personnaliser la génération des PDF présentait plusieurs risques, notamment des erreurs de syntaxe ou des modifications non intentionnelles. Pour répondre à cette problématique et faciliter l'utilisation par le personnel administratif, nous avons développé une interface web utilisant le framework Flask.

Le choix de Flask s'est imposé pour sa légèreté et sa facilité d'intégration avec l'écosystème Python existant. L'application s'organise autour d'un fichier principal app.py qui gère les routes et la logique métier, utilisant les outils existants comme ConfigHandler.py depuis le dossier tools/ pour la gestion de la configuration. Les templates HTML sont stockés dans le dossier templates/, tandis que les ressources statiques (CSS, JavaScript, images) sont organisées dans le dossier static/.

L'interface s'appuie sur les technologies web standards : HTML5 pour la structure, CSS3 pour le style via le fichier style.css, et JavaScript natif (vanilla) pour l'interactivité côté client avec main.js. Bootstrap 5 assure une mise en page responsive et professionnelle. Pour les communications asynchrones entre le client et le serveur, nous utilisons l'API Fetch, ce qui permet de mettre à jour dynamiquement l'interface sans recharge de page.

La génération des documents repose sur des commandes Make, déclenchées par deux boutons principaux : "Generate PDF" et "Delete All". Le bouton "Generate PDF" permet de déclencher automatiquement l'ensemble du processus de génération (JSON, Typst et PDF) en une seule action. Pour plus de flexibilité, un menu déroulant associé à ce bouton offre la possibilité d'exécuter individuellement chaque étape : la génération des fichiers JSON (make json), la création des fichiers Typst (make typst), ou la compilation des PDF (make pdf). De même, le bouton "Delete All" permet un nettoyage complet, tandis que son menu déroulant propose des options plus granulaires : make clean-json pour nettoyer les fichiers JSON, make clean-typst pour les fichiers Typst, et make clean-pdf pour les fichiers PDF.

L'interface web, illustrée dans la Figure 22, propose un tableau de bord complet pour le suivi et la gestion de la génération des documents. L'interface principale, via l'onglet "Status Overview", affiche l'état en temps réel des différents fichiers (JSON, Typst, PDF) pour les deux formations (FISE et FISEA). La barre supérieure contient trois menus déroulants : un sélecteur de branche GitLab permettant de choisir et basculer entre les différentes versions des cours (lors du changement de branche, les fichiers de cours actuels sont automatiquement remplacés par ceux de la nouvelle branche), un menu "Generate PDF" offrant le choix entre une génération complète ou des étapes spécifiques (JSON, Typst ou PDF), et un menu "Delete All" proposant soit une suppression complète soit un nettoyage sélectif (fichiers JSON, Typst ou PDF uniquement). Pour chaque fichier généré (JSON et PDF), un bouton de téléchargement devient disponible, permettant aux utilisateurs de récupérer directement les documents finaux pour les deux formations.

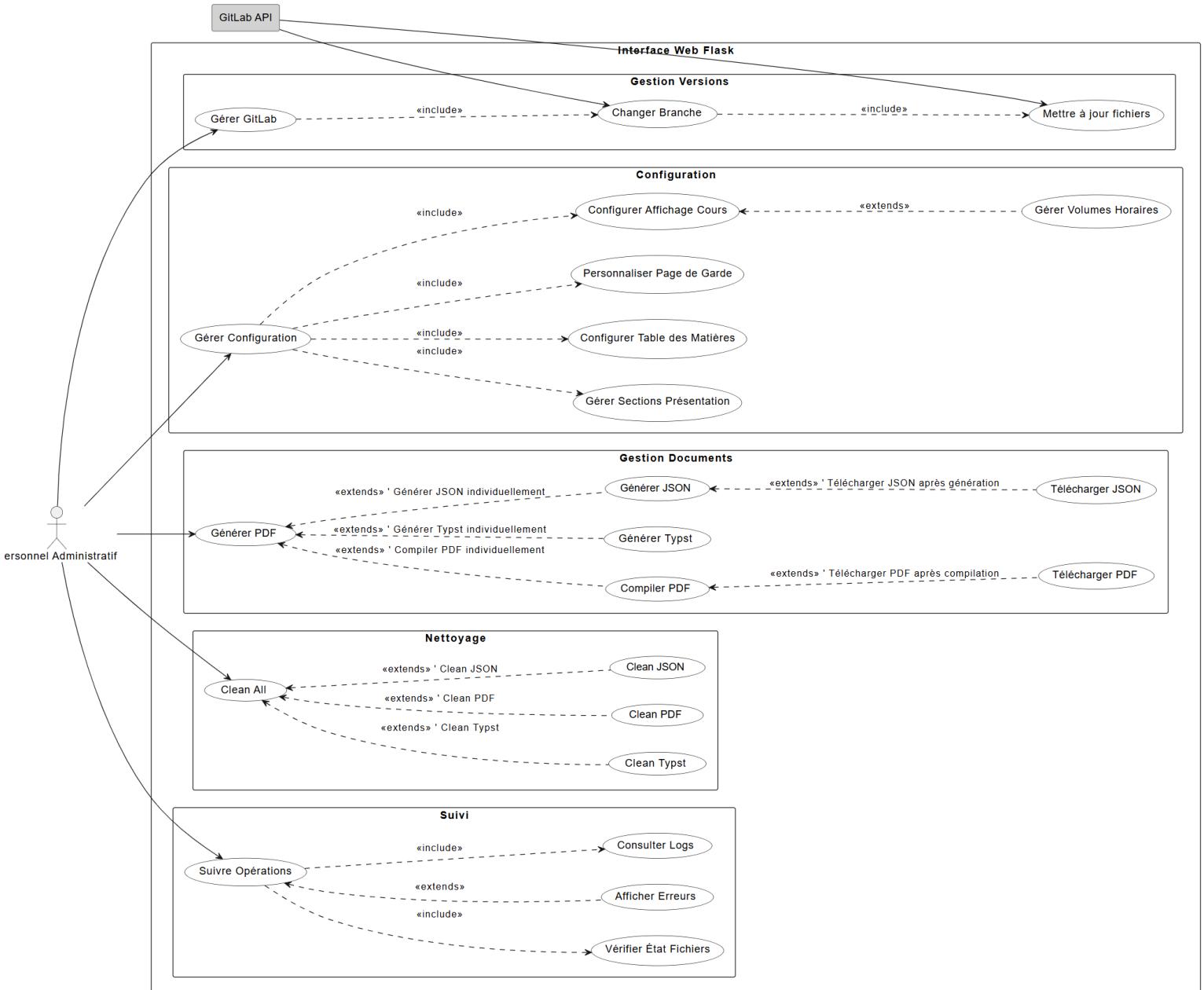
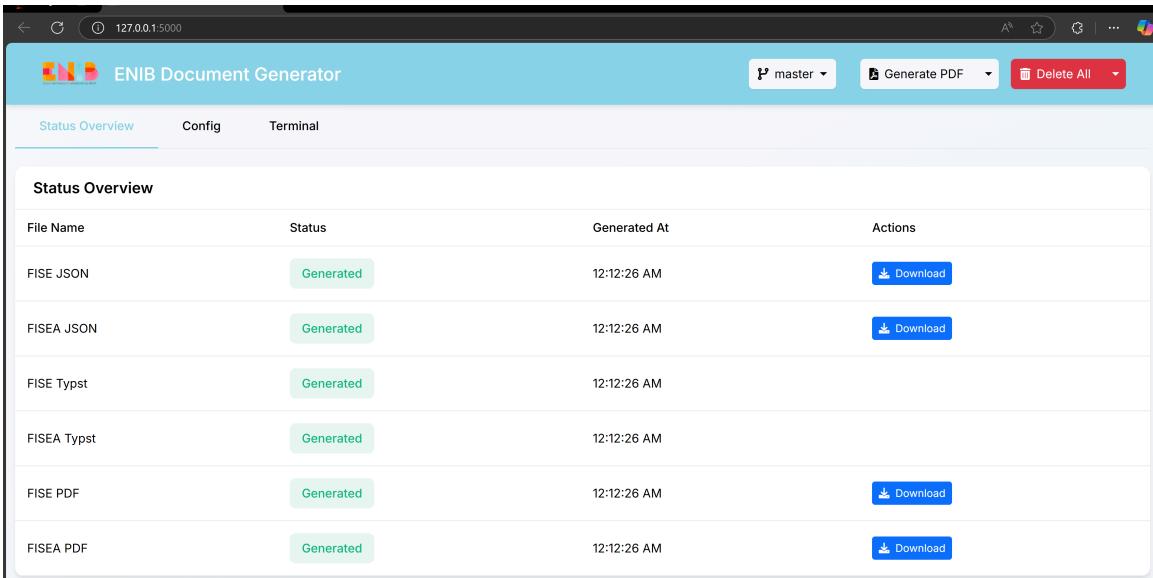


Figure 21: Diagramme de Cas d'Utilisation – Gestion et Génération de Documents PDF

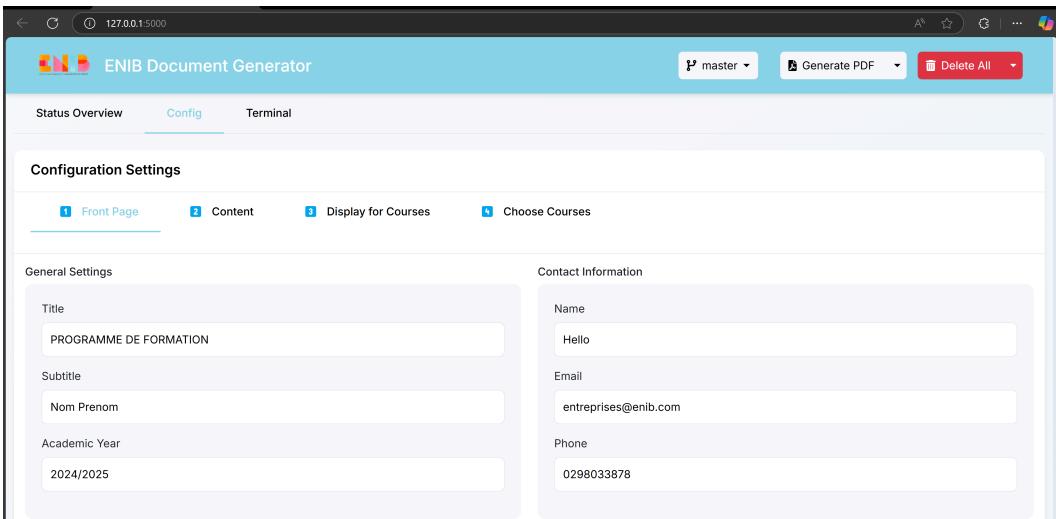


The screenshot shows the ENIB Document Generator interface with the "Status Overview" tab selected. It displays a table of generated files:

File Name	Status	Generated At	Actions
FISE JSON	Generated	12:12:26 AM	<button>Download</button>
FISEA JSON	Generated	12:12:26 AM	<button>Download</button>
FISE Typst	Generated	12:12:26 AM	
FISEA Typst	Generated	12:12:26 AM	
FISE PDF	Generated	12:12:26 AM	<button>Download</button>
FISEA PDF	Generated	12:12:26 AM	<button>Download</button>

Figure 22: Vue d'ensemble du tableau de bord avec l'état des fichiers générés

L'onglet "Config" s'organise en quatre sections distinctes. La section "Front Page" illustrée dans la Figure 23, permet la personnalisation de la page de garde, où l'utilisateur peut modifier le titre, le sous-titre, l'année académique et les informations de contact. La section "Content" illustrée dans la Figure 24 offre deux niveaux de personnalisation : la structure de la table des matières (affichage des semestres seuls ou avec leurs cours) et la sélection des sections de présentation du cursus (Organisation, Approche par Compétences, Intersemestres, sessions Automne-Printemps, Recrutement bac+2, Année 4 et 5).



The screenshot shows the "Config" tab with the "Front Page" section selected. It includes tabs for "Front Page", "Content", "Display for Courses", and "Choose Courses". The "Front Page" section contains fields for "Title" (PROGRAMME DE FORMATION), "Subtitle" (Nom Prenom), and "Academic Year" (2024/2025). The "Contact Information" section contains fields for "Name" (Hello), "Email" (entreprises@enib.com), and "Phone" (0298033878).

Figure 23: Configuration de la page de garde

Figure 24: Configuration des sections de contenu du document

La section "Display for Courses" illustrée dans la Figure 25, permet de définir finement l'affichage des informations. L'utilisateur peut choisir d'afficher uniquement les semestres, uniquement les cours, ou les deux. Pour chaque cours, il peut activer ou désactiver l'affichage du tableau d'informations générales, du tableau des volumes et du graphique AAT. Le tableau d'informations générales peut être personnalisé en sélectionnant les colonnes à afficher (UE, Type, Coefficient, Volume Estimé, Volume EDT).

Figure 25: Paramètres d'affichage des informations de cours

La dernière section, "Choose Courses", illustrée dans la Figure 26 présente une interface de sélection arborescente des cours organisée par semestre. Pour chaque semestre, les cours sont listés avec leur code, identifiant UE et volume horaire EDT actuel. L'interface permet de sélectionner individuellement chaque cours en cochant les cases correspondantes, et offre la possibilité de modifier directement le volume EDT pour chaque cours via un champ éditable. Toute modification du volume horaire est automatiquement enregistrée dans le fichier de configuration et se reflète dans le PDF généré.

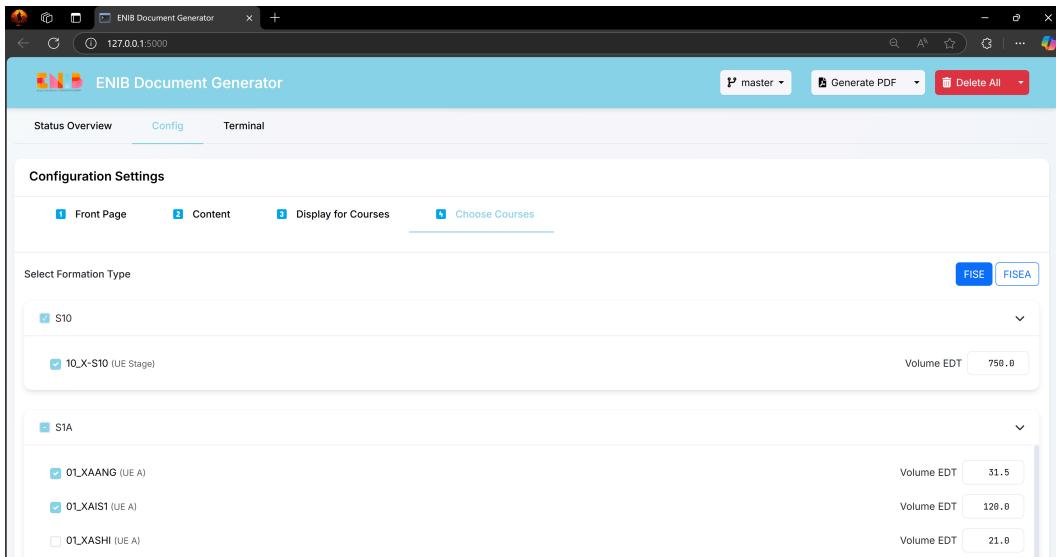


Figure 26: Interface de sélection des cours et gestion des volumes horaires

L'intégration avec GitLab constitue une amélioration importante de l'interface, permettant une gestion efficace des différentes versions des cours. Initialement, nous avions opté pour une approche basée sur Git, en effectuant un clone complet du projet dans un dossier temporaire, puis en copiant uniquement le dossier contenant les cours vers l'emplacement de travail. Cependant, cette méthode a montré ses limites lors des changements de branche, générant des conflits et des erreurs de gestion Git.

Pour résoudre ces problèmes, nous avons adopté une solution plus robuste utilisant l'API GitLab. Grâce à un token d'accès configuré dans GitLab, l'interface peut désormais récupérer directement les fichiers Markdown des cours depuis n'importe quelle branche du dépôt. Lorsqu'un utilisateur change de branche via le sélecteur, le système supprime automatiquement les fichiers de cours existants et télécharge ceux de la nouvelle branche sélectionnée. Cette approche simplifiée évite les complications liées à la gestion Git tout en permettant de générer des PDF basés sur différentes versions du programme pédagogique.

Pour éviter que l'interface ne télécharge systématiquement les fichiers depuis GitLab à chaque lancement, nous avons ajouté une étape de confirmation. Au démarrage de l'application, l'utilisateur est invité via le terminal à confirmer s'il souhaite mettre à jour les

fichiers depuis GitLab. Cette simple question permet d'optimiser le temps de démarrage en évitant des téléchargements inutiles lorsque les fichiers locaux sont déjà à jour.

Du point de vue technique, l'interaction côté client a été entièrement développée en JavaScript vanilla pour maintenir la simplicité et les performances de l'application. Les communications avec le serveur sont gérées via l'API Fetch, offrant une alternative moderne et performante à AJAX pour toutes les opérations asynchrones : sauvegarde de la configuration, récupération des statuts, et nettoyage des logs. Le suivi en temps réel de la génération des documents est assuré par un système d'indicateurs visuels dynamiques, où chaque type de fichier (JSON, Typst, PDF) dispose d'un badge de statut qui se met à jour automatiquement. Une console de logs interactive, utilisant un code couleur selon le niveau des messages (information, avertissement, erreur), permet aux utilisateurs de suivre précisément l'avancement des opérations et d'identifier rapidement les éventuels problèmes. L'ensemble de ces fonctionnalités est implémenté sans dépendance à des frameworks JavaScript complexes, garantissant ainsi des performances optimales tout en maintenant une grande maintenabilité du code.

5.3 Intégration FISEA dans VitePress

L'extension du système documentaire pour inclure la formation FISEA a nécessité une refonte de l'architecture existante. Ce développement s'est effectué en parallèle de la fusion des deux projets précédemment séparés (site VitePress et génération PDF) pour établir une véritable source unique de vérité. Cette section détaille les modifications techniques réalisées pour intégrer FISEA : adaptation des scripts de génération, mise à jour de la configuration VitePress et création des nouveaux fichiers de données.

5.3.1 Contexte et état initial

Le site statique VitePress était initialement conçu pour ne prendre en compte que la formation FISE. Le script `build_json.py` générait le fichier `aat_tree.json` pour la gestion des acquis d'apprentissage, tandis que `build_md.py` créait les index Markdown par semestre. VitePress utilisait ces fichiers pour construire le site web, en s'appuyant sur les fichiers `config.js` et `MenusConfig.js` placés dans le dossier `.vitepress` qui définissaient la navigation (navbar, sidebar) uniquement pour les semestres de la formation FISE. La génération des fichiers LaTeX était assurée par `build_tex.py` qui créait un fichier LaTeX pour chaque cours ainsi qu'un fichier principal `main.tex`. Lors de la compilation de `main.tex`, un PDF de 441 pages était généré, mais présentait plusieurs problèmes : table des matières incorrecte et mise en page incohérente.

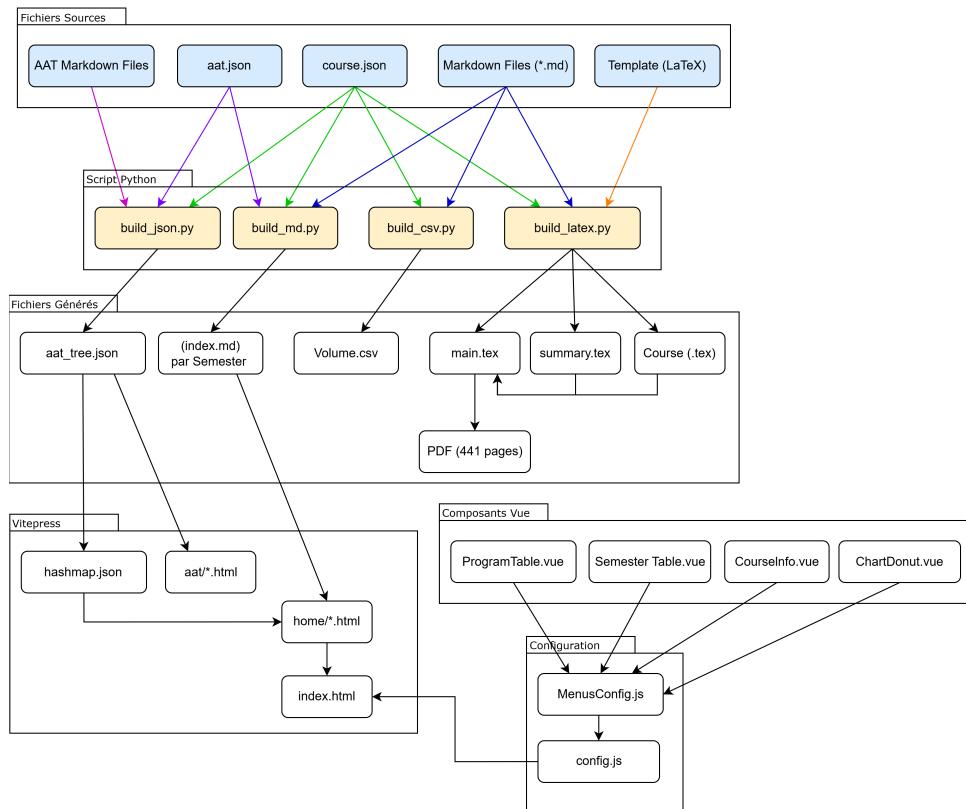


Figure 27: Pipeline de Génération VitePress et PDF (Avant Intégration FISEA)

L'architecture du système présentait deux contraintes majeures pour l'ajout de la formation FISEA. D'une part, la structure des données était conçue spécifiquement pour FISE, avec une arborescence reflétant uniquement les semestres de cette formation (S1A, S2P, S3A, etc.). D'autre part, le processus de génération PDF basé sur LaTeX ne permettait pas de créer facilement des documents personnalisés avec une sélection précise de semestres ou de cours, une fonctionnalité essentielle notamment dans le cadre des documents destinés aux OPCO ou d'autres cas.

5.3.2 Fusion des projets et objectifs FISEA

Avant d'intégrer FISEA, la première étape a consisté à fusionner le projet VitePress et celle de la génération de PDF, initialement distincts. Cette fusion visait à établir une source unique de vérité où les mêmes fichiers de cours Markdown serviraient à la fois au site web et aux documents PDF, éliminant les risques de divergence de contenu.

Les fichiers de cours pour FISE étaient initialement stockés dans docs/content et leur version anglaise dans docs/en. Pour maintenir cette organisation, les fichiers FISEA ont été ajoutés aux mêmes emplacements. La version anglaise des fichiers FISEA, qui n'existe pas initialement, a été créée par traduction des fichiers Markdown pour assurer une cohérence linguistique avec FISE.

L'ancien script `build_latex.py` a été retiré au profit de Typst pour la génération des PDF. Cette transition a permis de générer des PDF mieux structurés, sans les limitations rencontrées avec LaTeX. Les scripts existants (`build_csv.py`, `build_json.py`, `build_md.py`) sont demeurés chargés de l'extraction et du prétraitement des métadonnées, mais la production finale de PDF est déléguée à un nouveau flux utilisant Typst.

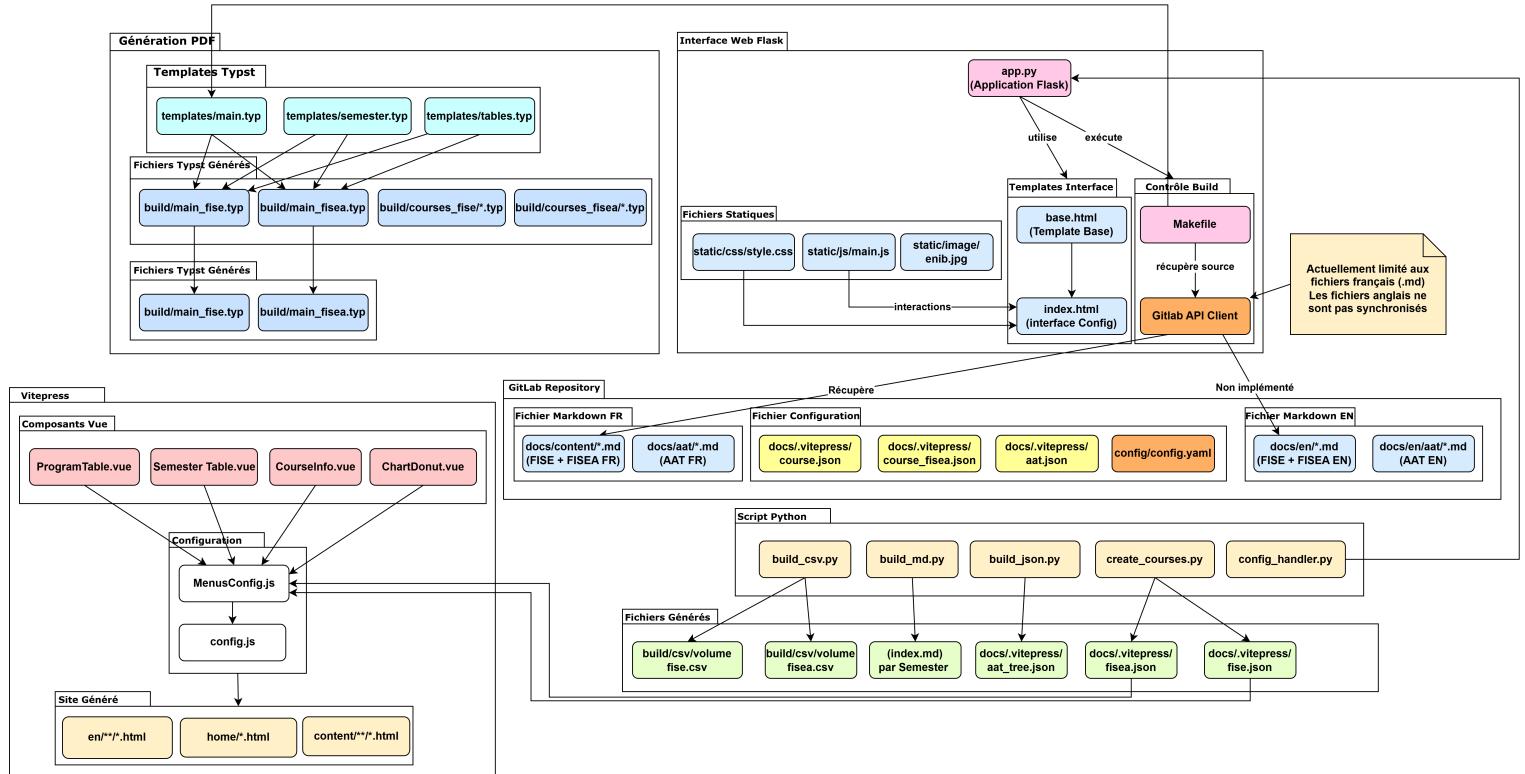


Figure 28: Pipeline de Génération VitePress et PDF (Après Intégration FISEA)

5.3.3 Mise à jour des scripts JSON et CSV

Les scripts de gestion des JSON et CSV ont dû être enrichis pour traiter les deux formations. Les scripts détectent désormais deux formations distinctes par la présence de `fise.json` et `fisea.json`. L'extraction des cours s'effectue dans chaque fichier, et deux séries de données peuvent être générées, par exemple `fise_volumes.csv` et `fisea_volumes.csv`. Cette séparation s'avère pratique car chaque formation apparaît avec ses propres semestres, volumes horaires et informations de cours.

Dans `build_csv.py`, la fonction `extract_course_data` a été adaptée pour prendre en compte la clé "FISE" ou "FISEA" et ainsi différencier les cheminements. Le script lit deux JSON distincts, puis écrit deux CSV, un pour chaque formation. L'ancien code n'extrayait que `course.json`, mais il s'agit maintenant d'englober `fise.json` et `fisea.json`, rendant le système plus souple sans rompre l'existant.

Le script build_json.py n'a nécessité que peu de modifications, hormis la possibilité de l'appeler deux fois (une pour course.json, une pour course_fisea.json) ou d'unifier la gestion AAT. L'essentiel tient au fait que la structure de la formation FISEA peut différer avec ses semestres alternés et cours spécifiques, mais reste manipulée par les mêmes routines JSON.

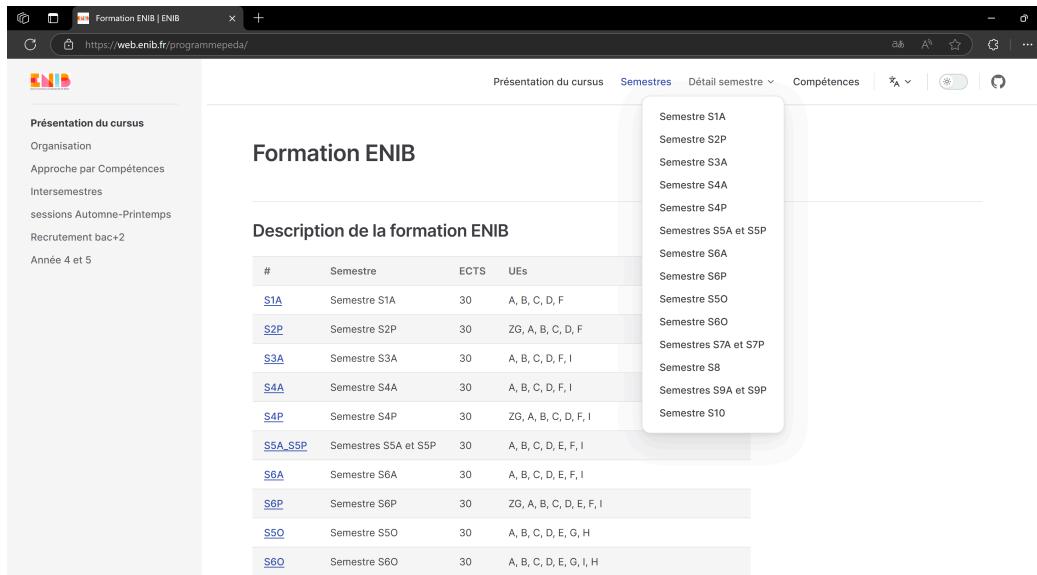
Le script build_md.py, qui crée les index par semestre, a été adapté pour générer non seulement les dossiers S1A, S2P, etc., mais aussi S5_FISEA, S6_FISEA. La nouveauté est l'ajout d'un paramètre program_path (dont la valeur peut être "fise" ou "fisea"), transmis aux fonctions export_index et sync_course_header. Il insère automatiquement un sous-répertoire fisea dans le chemin, aboutissant par exemple à ..//docs/content/fisea/S5_FISEA. Ce mécanisme permet de séparer physiquement les contenus Markdown des deux formations tout en réutilisant exactement les mêmes fonctions de génération.

5.3.4 Adaptation de la configuration VitePress

5.3.4.1 Configuration initiale

La configuration de VitePress, contenue dans config.js et MenusConfig.js, était focalisée sur la formation FISE. Les semestres mentionnés se bornaient à S1A, S2P, S3A, S4A, etc.

Les items de la sidebar et de la navbar redirigeaient l'utilisateur vers les dossiers correspondants (par exemple /content/S1A/), en s'appuyant sur un JSON unique, course.json.



The screenshot shows a web browser window for 'Formation ENIB | ENIB' at the URL <https://web.enib.fr/programmepeda/>. The main content area displays the 'Formation ENIB' page with a title 'Formation ENIB' and a section 'Description de la formation ENIB'. Below this is a table showing semester details:

#	Semestre	ECTS	UEs
S1A	Semestre S1A	30	A, B, C, D, F
S2P	Semestre S2P	30	ZG, A, B, C, D, F
S3A	Semestre S3A	30	A, B, C, D, F, I
S4A	Semestre S4A	30	A, B, C, D, F, I
S4P	Semestre S4P	30	ZG, A, B, C, D, F, I
S5A_S5P	Semestres S5A et S5P	30	A, B, C, D, E, F, I
S6A	Semestre S6A	30	A, B, C, D, E, F, I
S6P	Semestre S6P	30	ZG, A, B, C, D, E, F, I
S5O	Semestre S5O	30	A, B, C, D, E, G, H
S6O	Semestre S6O	30	A, B, C, D, E, G, I, H

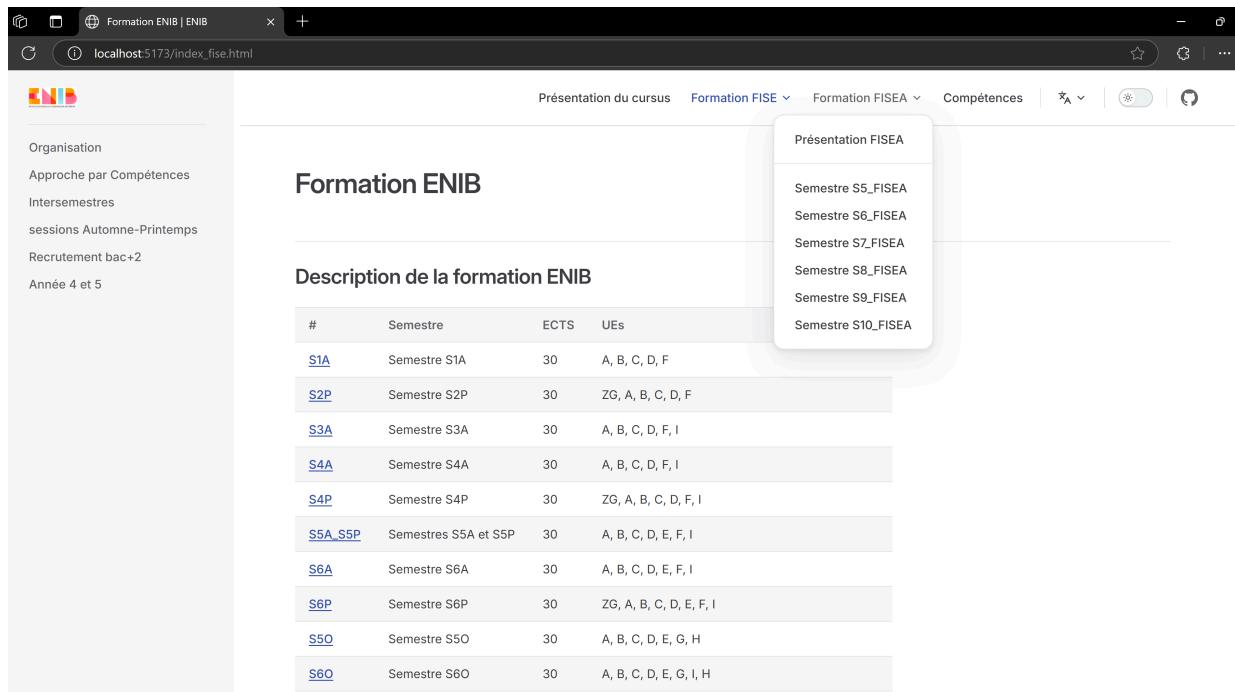
The sidebar on the left lists navigation items: Présentation du cursus, Organisation, Approche par Compétences, Intersemestres, sessions Automne-Printemps, Recrutement bac+2, and Année 4 et 5. The top navigation bar includes links for Présentation du cursus, Semestres, Détail semestre, Compétences, and other site navigation.

Figure 29: Interface initiale VitePress affichant uniquement les semestres FISE

5.3.4.2 Nouvelle configuration double formation

Deux nouveaux JSON (course_fise.json et éventuellement course_fise.en.json) ont été introduits, et MenusConfig.js importe désormais course_fisea.json et course_fisea_en.json. Les fonctions fiseaSemesterItems et enFiseaSemesterItems sont apparues, listant les semestres

S5_FISEA, S6_FISEA et ainsi de suite. Dans config.js, on combine les sidebars FISE et FISEA à l'aide de getSidebarSemester('fr', 'fise') et getSidebarSemester('fr', 'fisea'). On dispose aussi de pages d'accueil distinctes (index_fise.html, index_fisea.html) pour diriger l'utilisateur selon la formation choisie. VitePress affiche alors deux parcours distincts, correspondant à chacune des formations. L'utilisateur peut aisément naviguer entre les semestres FISE ou FISEA, et basculer en anglais si le site dispose des versions .en.



The screenshot shows a browser window for 'Formation ENIB | ENIB'. The URL is 'localhost:5173/index_fise.html'. The page title is 'Formation ENIB'. The top navigation bar includes links for 'Présentation du cursus', 'Formation FISE', 'Formation FISEA', and 'Compétences'. A dropdown menu for 'Formation FISEA' is open, listing semesters from 'Semestre S5_FISEA' to 'Semestre S10_FISEA'. The main content area is titled 'Formation ENIB' and 'Description de la formation ENIB'. Below this is a table listing semesters from S1A to S6O, each with its number, semester name, ECTS value (30), and a list of letters (A, B, C, D, F, G, H, I). The table rows are color-coded in pairs (light grey and white).

#	Semestre	ECTS	UEs
S1A	Semestre S1A	30	A, B, C, D, F
S2P	Semestre S2P	30	ZG, A, B, C, D, F
S3A	Semestre S3A	30	A, B, C, D, F, I
S4A	Semestre S4A	30	A, B, C, D, F, I
S4P	Semestre S4P	30	ZG, A, B, C, D, F, I
S5A_S5P	Semestres S5A et S5P	30	A, B, C, D, E, F, I
S6A	Semestre S6A	30	A, B, C, D, E, F, I
S6P	Semestre S6P	30	ZG, A, B, C, D, E, F, I
S5O	Semestre S5O	30	A, B, C, D, E, G, H
S6O	Semestre S6O	30	A, B, C, D, E, G, I, H

Figure 30: Nouvelle interface VitePress intégrant les semestres FISE et FISEA

5.3.5 Bénéfices de l'intégration

Cette intégration permet de gérer simultanément la formation FISE, qui existait déjà, et la formation FISEA, qui vient s'y ajouter, sans dupliquer entièrement les scripts ou altérer en profondeur la structure initiale. L'approche rend la maintenance plus aisée, puisque tout demeure centralisé dans une seule base, organisée par formation. La possibilité d'évoluer vers une troisième formation se trouve renforcée : il suffirait d'ajouter un nouveau JSON et quelques ajustements similaires pour étendre le site à un nouvel ensemble de cours et de semestres. Le passage de LaTeX à Typst modernise par ailleurs le flux de génération PDF et supprime la dépendance au script build_latex.py, tout en s'insérant harmonieusement dans la logique de scripts existants qui ne font que préparer les données.

L'intégration de FISEA dans VitePress a considérablement étendu le périmètre initial du site, conçu pour FISE uniquement. Les scripts Python gèrent désormais deux formations parallèlement et produisent JSON, CSV ou index Markdown spécifiques, tandis que VitePress offre deux parcours de navigation distincts. Les semestres FISE et FISEA cohabitent dans un

écosystème unifié, la maintenance restant concentrée sur des fichiers de configuration et de script uniques, aisément adaptables si de nouvelles évolutions apparaissent. La transition de LaTeX vers Typst assure un processus de génération PDF plus léger, tout en tirant parti de la structure de code existante. L'école dispose donc d'un système capable de gérer deux cursus distincts, voire davantage à l'avenir, tout en préservant la robustesse et la simplicité de la solution VitePress.

Conclusion

Le développement des interfaces utilisateur représente la dernière étape technique de notre projet de modernisation du système documentaire de l'ENIB. Les outils mis en place offrent une solution robuste et intuitive pour la gestion du programme pédagogique, en priorisant l'expérience utilisateur et l'efficacité opérationnelle.

L'interface web Flask, couplée à l'outil de gestion des métadonnées et à l'intégration FISEA dans VitePress, forme un écosystème cohérent qui répond pleinement aux besoins identifiés lors de l'analyse initiale, notamment pour la génération de documents adaptés aux exigences des différents acteurs internes et externes de l'école.

Le système mis en place démontre la pertinence des choix technologiques effectués. La migration vers Typst, l'utilisation de Flask pour l'interface web, et l'intégration réussie dans l'infrastructure GitLab existante ont permis de créer une solution performante et évolutive.

La prochaine étape du projet consistera à évaluer l'impact environnemental et sociétal de la solution mise en place, notamment en termes d'efficacité énergétique et d'accessibilité. Cette analyse RSE-RNE permettra de valider nos choix techniques et d'identifier d'éventuelles pistes d'amélioration pour réduire l'empreinte écologique du système.

CHAPITRE 6 : Étude d'impact RSE-RNE du projet

Introduction

Dans le contexte actuel de transformation numérique, la responsabilité des organisations s'étend au-delà des aspects sociaux traditionnels pour inclure une dimension numérique croissante. L'ENIB place le développement durable et la responsabilité sociétale (RSE), ainsi que la responsabilité numérique (RNE), au centre de sa stratégie [20].

Le projet de modernisation du système de gestion documentaire s'inscrit pleinement dans cette dynamique. Il vise à rationaliser la production de documents pédagogiques, à réduire l'empreinte environnementale et à favoriser une gouvernance numérique plus responsable. Ce chapitre explore en profondeur les impacts RSE-RNE de notre solution, en examinant à la fois les aspects techniques et organisationnels qui contribuent à cette démarche.

6 Gouvernance et organisation responsable

La gouvernance numérique responsable constitue un pilier essentiel du projet. L'ENIB cherche à instaurer des pratiques de gestion documentaire transparentes et éthiques, en s'appuyant sur des outils modernes et des processus rigoureux.

La centralisation des données pédagogiques repose sur des fichiers Markdown stockés dans GitLab[21]. Ces fichiers constituent la "single source of truth" pour deux systèmes distincts : VitePress pour la génération du site web pédagogique et Typst pour la génération de documents PDF. Cette approche évite la duplication des contenus et réduit les risques d'incohérence, tout en répondant aux exigences de sobriété numérique.

Chaque modification des fichiers Markdown est soumise à un processus de validation rigoureux via GitLab[21]. L'ingénierie pédagogique valide systématiquement les contributions des enseignants et administratifs avant leur intégration, garantissant ainsi la qualité et la cohérence des documents produits. GitLab permet également une traçabilité complète des modifications, assurant une transparence totale sur l'évolution des documents.

L'interface web développée avec Flask joue un rôle clé dans la gouvernance participative du projet. Elle permet aux utilisateurs de personnaliser les documents PDF sans manipuler directement les fichiers source, facilitant ainsi la contribution des différents acteurs de l'école. Cette approche réduit la charge cognitive des utilisateurs et améliore leur implication dans la gestion documentaire[22].

6.1 Impact environnemental et sobriété numérique

Le projet adopte une approche de sobriété numérique à travers des choix technologiques visant à optimiser les ressources et à réduire l'empreinte environnementale.

L'un des principaux leviers d'optimisation réside dans le remplacement de LaTeX par Typst pour la génération de documents PDF. Typst est plus rapide et consomme moins de ressources CPU que LaTeX, réduisant ainsi la consommation électrique des serveurs. De plus, Typst nécessite une installation d'environ 100 Mo, contre plus de 3 Go pour LaTeX, ce qui allège considérablement l'empreinte numérique[23].

L'environnement de développement est standardisé grâce à Miniconda, qui permet de créer des environnements virtuels isolés. En utilisant un fichier typst_environment.yml, il est possible de reproduire l'environnement de génération de PDF sur n'importe quelle machine disposant de Miniconda, sans nécessiter d'installation manuelle supplémentaire. Cette approche évite la multiplication d'installations inutiles et garantit la reproductibilité du système de manière légère et écoresponsable.

Pour la partie web, VitePress nécessite quelques installations supplémentaires comme gray-matter, Chart.js, vite, et vue. Contrairement à la génération PDF, VitePress dépend de Node.js, ce qui implique des installations spécifiques en l'absence de ce dernier. Toutefois, ces installations sont limitées et n'alourdissent pas excessivement l'environnement numérique de l'école.

La génération de documents PDF est réalisée uniquement à la demande, évitant ainsi la production systématique de fichiers inutiles. Ce mode de fonctionnement permet de limiter l'utilisation des ressources serveurs, tout en répondant de manière précise aux besoins spécifiques.

L'interface web développée avec Flask en Python s'inscrit dans cette logique de sobriété numérique. Contrairement aux solutions client-lourdes comme Qt, Flask permet de s'appuyer directement sur le navigateur, évitant ainsi des installations locales lourdes et réduisant l'empreinte numérique. Cette approche légère et multiplateforme simplifie l'accès au système, tout en garantissant une meilleure portabilité et une maintenance facilitée[24].

La génération PDF via Typst est déclenchée uniquement lorsque nécessaire, optimisant l'utilisation des ressources serveurs et réduisant la consommation énergétique. En adoptant Flask et Typst, l'ENIB priviliege des solutions efficaces et durables, en adéquation avec ses engagements de sobriété numérique.

6.2 Relations de travail et inclusion numérique

Le projet contribue à améliorer les conditions de travail du personnel administratif en automatisant les tâches répétitives liées à la génération et à la mise à jour des documents

pédagogiques. Cette automatisation libère du temps pour des activités à plus forte valeur ajoutée, telles que l'accompagnement des étudiants ou l'amélioration des processus pédagogiques.

Exemple concret d'automatisation : La création de livrets pédagogiques et de programmes de cours est désormais automatisée. Auparavant, ces documents étaient élaborés manuellement, un processus long et fastidieux nécessitant souvent plusieurs heures de travail. Dans certains cas, deux personnes étaient mobilisées pour rédiger, vérifier et mettre en page ces livrables afin de garantir leur conformité et leur qualité. Cette charge de travail ralentissait la production et augmentait le risque d'erreurs ou d'incohérences[25].

Désormais, grâce à l'intégration de Typst, les informations saisies dans les fichiers Markdown sont automatiquement formatées et compilées en documents prêts à l'emploi. Ce processus garantit une mise en page uniforme, réduit considérablement le temps de création et libère les équipes de tâches répétitives, leur permettant ainsi de se concentrer sur des missions à plus forte valeur ajoutée.

L'interface web développée permet à tous les utilisateurs, quel que soit leur niveau technique, de générer des documents PDF de manière intuitive. La simplicité de cette interface réduit la charge cognitive et favorise une plus grande inclusion numérique au sein de l'école.

Le site web VitePress permet de rendre les programmes pédagogiques accessibles en français et en anglais, renforçant ainsi l'inclusion des étudiants et enseignants internationaux. Cette fonctionnalité contribue à la diversité linguistique de l'école et facilite la communication avec les partenaires étrangers.

6.3 Impact sociétal et développement numérique local

Le projet exerce un impact significatif sur l'écosystème éducatif et local. En centralisant et standardisant la documentation pédagogique, il rend l'information plus accessible et compréhensible pour les étudiants, les familles et les entreprises partenaires.

Dans le cadre de ce projet, un template de rapport de stage a été développé en utilisant Typst. Ce template, conforme aux exigences de l'ENIB, sera mis à disposition des futurs étudiants pour leur permettre de rédiger leurs rapports de manière structurée et professionnelle. Ce choix s'inscrit dans une démarche d'initiation des étudiants aux outils numériques modernes [25].

Historiquement, un template LaTeX, conçu par M. Vincent Choqueuse, était proposé aux étudiants. Toutefois, LaTeX présente une courbe d'apprentissage relativement élevée et une configuration complexe, ce qui pouvait décourager certains utilisateurs. Typst, en revanche, se distingue par sa simplicité syntaxique et sa rapidité de compilation, offrant une alternative plus accessible et moins gourmande en ressources.

L'utilisation de ce nouveau template ne se limite pas à alléger la charge de travail des étudiants. Il favorise également leur acculturation aux technologies émergentes, les préparant ainsi à adopter des outils modernes dans leur futur environnement professionnel. De plus, cette initiative contribue à renforcer les compétences numériques des étudiants, un atout précieux dans un marché du travail en constante évolution.

Le projet n'impacte pas uniquement les étudiants, mais bénéficie également à l'écosystème académique local. En simplifiant la gestion documentaire et en standardisant les formats utilisés, l'ENIB améliore sa réactivité face aux demandes des entreprises partenaires. Les dossiers d'alternance, les programmes de formation et les brochures institutionnelles peuvent être produits plus rapidement, avec une qualité constante. Cette efficacité accrue favorise l'an-crage territorial de l'école et renforce son attractivité auprès des acteurs économiques locaux.

6.4 Comparaison avec d'autres établissements

L'ENIB s'inscrit dans une dynamique partagée par plusieurs établissements d'enseignement supérieur, visant à moderniser les processus de gestion documentaire grâce à des outils numériques plus légers et respectueux de l'environnement.

Dans certaines écoles, l'usage de LaTeX reste la solution privilégiée pour la production de documents académiques. Par exemple, l'INSA Lyon propose des templates LaTeX dédiés à la rédaction de thèses et mémoires, garantissant une standardisation des livrables et répondant aux exigences académiques[26], [27].

Cependant, malgré la robustesse de LaTeX, ses limites en termes de lourdeur, de complexité et de consommation de ressources ont conduit l'ENIB à opter pour Typst. Ce dernier permet une génération de documents PDF plus rapide, légère et facile à maintenir, contribuant ainsi à la réduction de l'empreinte environnementale de l'établissement. Contrairement à LaTeX, Typst s'intègre de manière plus fluide avec les autres outils déjà en place, tels que VitePress et Flask, créant ainsi un écosystème cohérent et performant.

L'approche adoptée par l'ENIB illustre une volonté d'innovation continue, tout en s'alignant avec des objectifs de sobriété numérique et de responsabilité sociétale. Cette stratégie positionne l'école à la pointe des pratiques de transformation numérique durable dans l'enseignement supérieur.

6.5 Impacts indirects et effets numériques induits

Au-delà des bénéfices directs observés, la modernisation du système documentaire génère des impacts indirects significatifs. En introduisant des outils modernes comme Typst

et VitePress, l'ENIB contribue au développement des compétences numériques de ses enseignants et personnels administratifs.

L'utilisation quotidienne de GitLab pour gérer les fichiers Markdown et suivre les modifications favorise l'appropriation de pratiques de versionnage, essentielles dans les métiers du numérique. Ce type d'outil, initialement limité à la gestion documentaire, pourrait être étendu à d'autres projets internes, renforçant ainsi la transformation numérique globale de l'établissement.

L'automatisation de la génération de documents améliore également la capacité de l'ENIB à répondre rapidement aux demandes externes. Par exemple, les demandes de documents pédagogiques spécifiques formulées par les OPCO (Opérateurs de Compétences) dans le cadre des formations en alternance peuvent désormais être satisfaites plus efficacement, grâce à des outils qui permettent une personnalisation rapide et précise des contenus.

De plus, l'architecture modulaire du système facilite son extension à d'autres domaines de l'école. Les composants développés pour la génération des PDF ou l'interface de gestion documentaire pourraient être réutilisés pour d'autres projets, accélérant ainsi la digitalisation de l'ensemble des services administratifs.

6.6 Perspectives d'amélioration RSE-RNE

Bien que les résultats obtenus soient encourageants, plusieurs pistes d'amélioration ont été identifiées pour renforcer encore davantage l'impact RSE-RNE du projet.

Sur le plan technique, la mise en place d'un système de monitoring des ressources serveurs pourrait permettre de mesurer précisément l'empreinte carbone associée à la génération des documents PDF. Cette mesure offrirait une vision claire des consommations énergétiques et permettrait d'identifier des axes de réduction supplémentaires.

L'optimisation continue des dépendances du projet, notamment pour l'environnement VitePress, pourrait également contribuer à réduire l'empreinte numérique. L'exploration de solutions alternatives plus légères à certaines bibliothèques comme Chart.js ou gray-matter pourrait s'inscrire dans cette démarche.

6.7 Recommandations pour une évolution durable

Pour assurer la pérennité et l'amélioration continue de cette transformation numérique, il est essentiel de mettre en place des outils permettant de mesurer et réduire l'empreinte carbone du système. L'intégration de solutions telles que Scaphandre (outil de mesure en temps réel de la consommation énergétique des serveurs)[28] ou CO2.js (qui estime l'empreinte carbone des sites web)[29] permettrait de suivre précisément l'impact environnemental des opérations numériques, comme la génération de PDF avec Typst ou les déploiements VitePress.

En complément, des outils tels que Green Metrics Tool ou Cloud Carbon Footprint pourraient être utilisés pour analyser l'impact carbone des infrastructures cloud et services associés, offrant ainsi une vision globale, tant locale que dématérialisée. Cette approche garantirait une évaluation complète de l'ensemble du système, couvrant les serveurs sur site ainsi que les ressources hébergées dans le cloud.

Ces outils pourraient être automatisés et intégrés directement dans le pipeline GitLab CI/CD, générant des rapports d'empreinte carbone après chaque compilation ou mise à jour du site. Cette automatisation offrirait un suivi constant sans nécessiter d'intervention humaine, tout en sensibilisant les équipes à l'impact réel des opérations numériques.

L'ENIB pourrait également renforcer sa collaboration avec d'autres écoles engagées dans des démarches similaires, comme l'INSA Lyon ou l'Université de Nantes. Le partage des outils développés (comme les scripts Flask ou Typst) et la mutualisation des innovations techniques permettraient d'optimiser les ressources tout en accélérant les avancées technologiques.

Enfin, des développements futurs pourraient inclure l'ajout d'une génération PDF multilingue, actuellement limitée au site web VitePress, ainsi que l'amélioration de l'interface Flask pour rendre la personnalisation des documents plus intuitive. L'intégration d'un tableau de bord interactif, affichant en temps réel l'empreinte carbone du système grâce à Scaphandre ou CO2.js, offrirait une transparence accrue et renforcerait la sensibilisation à la sobriété numérique.

En mettant en œuvre ces actions, l'ENIB s'affirmera comme un acteur clé de la digitalisation responsable, réduisant son empreinte écologique tout en poursuivant son engagement vers une transformation numérique durable et performante.

6.8 Conclusion

Le projet de modernisation documentaire de l'ENIB représente une avancée significative vers une gestion plus responsable et durable des ressources pédagogiques. L'utilisation de Typst pour la génération des documents PDF, ainsi que la création d'un template de rapport de stage, illustrent une volonté claire d'optimiser les processus tout en préparant les étudiants aux technologies de demain.

En remplaçant des solutions plus lourdes comme LaTeX par des outils modernes et plus légers, l'ENIB démontre qu'il est possible de concilier innovation technologique, efficacité opérationnelle et responsabilité environnementale. Les choix technologiques et organisationnels effectués au cours de ce projet s'inscrivent dans une logique de transformation numérique durable, en harmonie avec les objectifs de développement durable de l'établissement. La mise à disposition du template Typst pour les futurs étudiants reflète cette volonté de partager et de diffuser des pratiques numériques responsables. En

s'engageant dans cette démarche, l'ENIB se positionne non seulement comme un acteur de la transition écologique, mais également comme un précurseur dans l'intégration des outils numériques dans les processus pédagogiques et administratifs. À travers cette initiative, l'ENIB s'affirme non seulement comme un acteur clé de la transition numérique responsable mais aussi comme une référence pour d'autres établissements souhaitant s'engager dans des démarches similaires.

7 Conclusion générale :

Ce projet de modernisation du système documentaire de l'ENIB marque une évolution majeure dans la gestion des documents pédagogiques de l'école. Face à un système initial fragmenté reposant sur LaTeX et des processus manuels chronophages, nous avons développé une solution répondant efficacement aux besoins de l'établissement.

L'architecture technique mise en place repose sur deux projets complémentaires qui ont été fusionnés pour garantir une source unique de vérité. Le premier projet, centré sur Typst et Flask, automatise la génération de documents PDF personnalisables. L'interface web développée permet au personnel administratif de produire rapidement des documents adaptés aux exigences des OPCO et autres partenaires. Le second projet enrichit le site web VitePress existant pour intégrer la formation FISEA. Bien que techniquement indépendants, ces deux systèmes s'appuient sur les mêmes fichiers Markdown sources centralisés sur GitLab, assurant ainsi une parfaite cohérence des informations diffusées.

Cette transformation s'inscrit pleinement dans la démarche RSE-RNE de l'ENIB. L'utilisation d'outils légers comme Typst réduit significativement l'empreinte environnementale du système, tandis que le développement d'un template de rapport de stage modernise les outils mis à disposition des étudiants.

Les perspectives d'évolution sont nombreuses : optimisation des performances, nouvelles fonctionnalités, renforcement du monitoring environnemental. La solution développée constitue un socle solide pour accompagner les futures transformations de l'ENIB, notamment son intégration au nouvel Institut National Polytechnique.

Ce projet démontre qu'une approche technique moderne, associée à une vision responsable, peut générer des solutions performantes qui améliorent concrètement le quotidien des utilisateurs. L'ENIB dispose désormais d'un système documentaire fiable et évolutif, aligné sur ses objectifs de transformation numérique durable.

BIBLIOGRAPHIE

- [1] "École Nationale d'Ingénieurs de Brest - Wikipédia." [Online]. Available: https://fr.wikipedia.org/wiki/%C3%89cole_nationale_d%27ing%C3%A9nieurs_de_Brest
- [2] "Développement Durable et Responsabilité Sociétale - ENIB." [Online]. Available: <https://www.enib.fr/dd-rse/>
- [3] "Centre Européen de Réalité Virtuelle (CERV) - ENIB." [Online]. Available: <https://cerv.enib.fr/>
- [4] "L'École Nationale d'Ingénieurs de Brest a inauguré La Forge, son laboratoire de fabrication." [Online]. Available: <https://www.letelegramme.fr/finistere/brest-29200/lecole-nationale-dingenieurs-de-brest-a-inaugure-la-forge-son-laboratoire-de-fabrication-6474744.php>
- [5] "Institut de Recherche Dupuy de Lôme (IRDL) - Présentation." [Online]. Available: <https://www.irdl.fr/nous-connaître/>
- [6] "Lab-STICC - ENIB." [Online]. Available: <https://www.enib.fr/recherche-transfert-technologique/labsticc-2/>
- [7] "Plaquette de Formation - ENIB." [Online]. Available: <https://web.enib.fr/files/plaquettes/formation-vf-web.pdf>
- [8] "Devenir Ingénieur - Cycle Ingénieur ENIB." [Online]. Available: <https://www.enib.fr/devenir-ingénieur/ingenieur-cycle-ingenieur>
- [9] "Formation Ingénieur Étudiant puis Apprenti (FISEA) - ENSEEIHT." [Online]. Available: <https://www.enseeiht.fr/fr/formation/formation-ingenieur-etudiant-puis-apprenti-fisea.html>
- [10] "Formations d'Ingénieurs en Alternance Interne - CTI." [Online]. Available: https://www.cti-commission.fr/wp-content/uploads/2023/07/Formations-Alternance-interne_CTI_19-04-2023.pdf
- [11] "Association Nationale des Ingénieurs de l'ENI de Brest (ANIENIB)." [Online]. Available: <https://anienib.fr/>
- [12] "Rencontrer nos étudiants - ENIB." [Online]. Available: <https://www.enib.fr/rencontrer-nos-etudiants/>
- [13] "Référentiel 2024 de la CTI." [Online]. Available: https://www.cti-commission.fr/wp-content/uploads/2024/02/RO_Référentiel_2024_VD.pdf
- [14] "Les Opérateurs de Compétences (OPCO)." [Online]. Available: <https://www.alternance-professionnelle.fr/opco/>
- [15] "Le Contrat d'Apprentissage - OPCO Mobilités." [Online]. Available: <https://www.opcomobilites.fr/entreprise/sinformer-sur-les-dispositifs/le-contrat-dapprentissage>
- [16] "Approche par Compétences et Fiche RNCP dans les Formations d'Ingénieurs." [Online]. Available: <https://www.cti-commission.fr/la-fiche-rncp-et-son-lien-avec-la-demarche-competences-au-sein-des-formations-dingenieurs-et-de-bachelor-en-sciences-et-ingenierie>
- [17] T. Team, "Typst: A Modern Document Formatting Language." [Online]. Available: <https://typst.app/>

- [18] L. Lamport, *LaTeX: A Document Preparation System*. Reading, Massachusetts: Addison-Wesley, 1986.
- [19] J. Doe, "Comparing Typst and LaTeX: A Modern Approach to Document Formatting," *Journal of Document Engineering*, vol. 41, no. 4, pp. 78–85, 2022, doi: 10.1234/jde.2022.0056.
- [20] "Développement Durable et Responsabilité Sociétale (RSE) - ENIB." [Online]. Available: <https://www.enib.fr/dd-rse/>
- [21] "Plateforme GitLab de l'ENIB." [Online]. Available: <https://git.enib.fr/>
- [22] "Responsabilité Numérique des Entreprises - Enjeux Environnementaux et Sociaux." [Online]. Available: <https://www.strategie.gouv.fr/infographies/responsabilite-numerique-entreprises-enjeux-donnees-environnementaux-sociaux>
- [23] "TYPST: A new markup + page layout engine to take on LaTeX." [Online]. Available: <https://forum.literatureandlatte.com/t/typst-a-new-markup-page-layout-engine-to-take-on-latex/134674>
- [24] "Why Should You Use Flask: 7 Reasons." [Online]. Available: <https://www.planeks.net/why-use-flask/>
- [25] "Template de Rapport de Stage - ENIB." [Online]. Available: https://git.enib.fr/y24terro/template_rapport_de_stage
- [26] "Modèle LaTeX de manuscrit - INSA Lyon." [Online]. Available: <https://fedorainsa-lyon.fr/fr/content/template-latex-de-manuscrit>
- [27] "INSA Rapport de stage Template sur Overleaf." [Online]. Available: <https://fr.overleaf.com/latex/templates/insa-rapport-de-stage-template/gvtyxbybyfzq>
- [28] "Scaphandre - Monitoring de la consommation énergétique des serveurs." [Online]. Available: <https://github.com/hubble-org/scaphandre>
- [29] "CO2.js - Estimation de l'empreinte carbone des sites web." [Online]. Available: <https://www.thegreenwebfoundation.org/co2-js/>

ANNEXES

https://git.enib.fr/a24machko/md_to_typst

<https://git.enib.fr/a24machko/vitepress>

https://git.enib.fr/y24terro/template_rapport_de_stage

https://git.enib.fr/y24terro/rapport_de_stage_2024_2025