

PROMO 2025-2026



TER

CAHIER DES CHARGES :

Projet JADE – Justice

Algorithmique des Élections

DJENNAD Mahdi

RABEHI Milhane

Master 1 MIASHS

M. RAMBAUD Romain (porteur du  
projet JADE (CRJ))

Mme. BLIGNY Caroline  
(informaticienne, cheffe de projet/  
experte en ingénierie logicielle (LJK))

## Table des matières :

<u>1.</u>	<u>INFORMATIONS PROJET</u>	2
1.A	PORTEURS DU PROJET .....	2
1.B	CADRE.....	2
1.C	PERIODE.....	2
<u>2.</u>	<u>INTRODUCTION ET PRÉSENTATION DU PROJET</u> .....	ERREUR ! SIGNET NON DEFINI.
2.A	CONTEXTE SCIENTIFIQUE : .....	2
2.B	LA PROBLEMATIQUE :.....	2
2.C	OBJECTIF DU STAGE :.....	2
<u>3.</u>	<u>ANALYSE DE L'EXISTANT</u> .....	3
<u>4.</u>	<u>SPECIFICATIONS FONCTIONNELLES ET OBJECTIFS</u> .....	4
4.A	ANALYSE ET MODELISATION (PRIORITE 1) .....	4
4.B	DEVELOPPEMENT DE L'OUTIL PYTHON.....	4
4.C	CONTROLE DE QUALITE ET FIABILISATION.....	4
<u>5.</u>	<u>RESSOURCES ET CONTRAINTES</u> .....	4
<u>6.</u>	<u>PLANNING ET MISE EN ŒUVRE (GANTT)</u> .....	5
<u>7.</u>	<u>ORGANISATION ET REPARTITION</u> .....	6
7.A	ROLES ET RESPONSABILITES .....	6
7.B	ÉQUILIBRE DE LA CHARGE (270H) .....	6

# 1. Informations projet

## 1.a Porteurs du projet

- DJENNAD Mahdi
- RABEHI Milhane
- Encadrement : Sous la direction de Madame Caroline Bligny (UGA).

## 1.b Cadre

Travail d'Étude et de Recherche (TER) – Master MIASHS (Mathématiques et Informatique Appliquées aux Sciences Humaines et Sociales) – Spécialité Informatique et Cognition. Volume horaire : 270 heures par étudiant.

## 1.c Période

Décembre 2025 – Juin 2026

# 2. Introduction et présentation du projet

## 2.a Contexte Scientifique :

Le projet JADE (Justice Algorithmique des Élections) est une initiative interdisciplinaire soutenue par l'IDEX de l'Université Grenoble Alpes. Il réunit des chercheurs en droit (CRJ) et en sciences de l'information (LJK/LIG) pour analyser le contentieux électoral de la Vème République à travers le prisme des mathématiques appliquées et de l'intelligence artificielle.

## 2.b La Problématique :

Si la base JADE consolide déjà les résultats électoraux et les métadonnées des décisions, elle ne contient pas encore les abstracts du Conseil constitutionnel. Ces résumés juridiques sont pourtant cruciaux : ils explicitent le raisonnement du juge et l'impact des irrégularités sur la sincérité du scrutin, ce que les données brutes ne permettent pas de saisir.

The screenshot shows a page from the Constitutional Council's website. At the top, it says "Décision n° 2013-4831 AN du 22 mars 2013". Below this, there are four links: "Abstract" (with a small document icon), "Commentaire" (with a small document icon), "Version PDF de la décision" (with a small document icon), and "Lien stable de la décision" (with a small link icon). To the right, under the heading "LE CONSEIL CONSTITUTIONNEL", there is a detailed summary of the decision, mentioning M. Hervé DROUOT and his campaign financing issues. At the bottom, it notes that observations were presented by M. DROUOT on February 26, 2013.

Figure 1 : Exemple d'un "Abstract" (analyse juridique) consultable sur le site du Conseil Constitutionnel (Décision n° 2013-4831 AN)

## 2.c Objectif du Stage :

L'enjeu de ce TER est de concevoir et déployer une chaîne de traitement automatisée sous Python. Cette solution devra assurer l'extraction, le parsing (analyse syntaxique) et l'intégration des abstracts (corpus allant jusqu'en 2023) au sein de l'architecture PostgreSQL existante, tout en garantissant l'intégrité des liens relationnels avec les décisions déjà répertoriées.

### 3. Analyse de l'existant

- Architecture de la Base de Données JADE :

  1. Le système repose sur une instance PostgreSQL structurée pour croiser données électorales et juridiques.
  2. La table centrale, public.decision, est déjà opérationnelle et contient les métadonnées critiques : id\_source (identifiant unique CONTEXT), numero (référence normalisée, ex: 2022-5800 AN), ecli, date\_decision et solution.
  3. Cette table sert de pivot pour l'ensemble du projet, mais elle ne dispose pas encore de lien vers le contenu textuel des résumés.

```
CREATE TABLE public.decision (
    id_decision integer NOT NULL,
    id_source text NOT NULL,
    numero text NOT NULL,
    date_dec date NOT NULL,
    ecli text NOT NULL,
    nature text NOT NULL,
    solution text,
    url_cc text,
    departements text,
    circonscriptions text,
    type_requete text,
    annee_saisine integer
);
```

Figure 2 : Structure SQL actuelle de la table pivot public.decision

- Audit des Sources "Abstracts" :

  1. Le corpus est actuellement fragmenté en trois fichiers distincts déposés sur le Cloud UGA.
  2. Ces fichiers contiennent l'historique des résumés jusqu'en 2023.
  3. L'enjeu technique identifié est la réconciliation de ces données éparses avec les entrées existantes de la table public.decision via un identifiant commun (le numéro de décision ou l'ECLI).

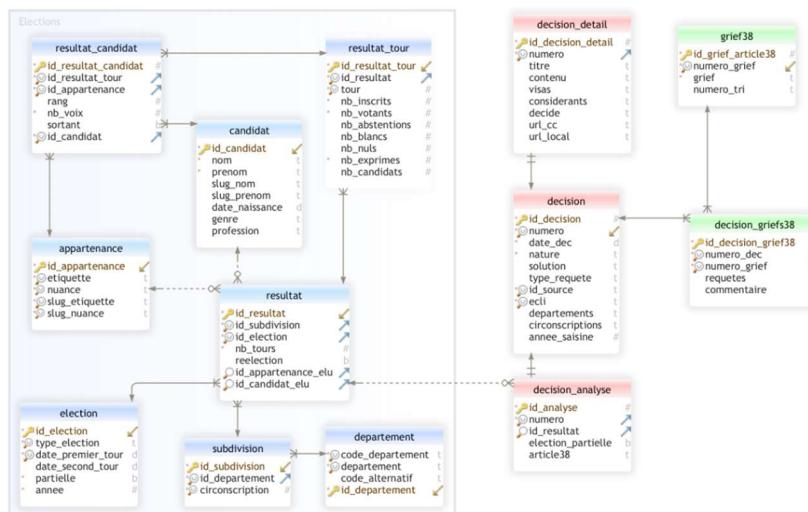


Figure 3 : Architecture globale de la base JADE et localisation des sources de données.

# 4. Spécifications Fonctionnelles et Objectifs

## 4.a Analyse et Modélisation (Priorité 1)

- Stratégie de "Staging" : Chargement des fichiers sources dans des tables "brutes" (staging area) sous PostgreSQL. Cette étape est cruciale pour auditer la qualité des données initiales et valider les cardinalités (relations 1:1 ou 1:N entre décisions et abstracts).
- Conception du Modèle Relationnel : Définition de la structure de la future table abstracts et des contraintes d'intégrité (clés étrangères) pour assurer un lien robuste avec la table public.decision.

## 4.b Développement de l'outil python

- Module d'Extraction : Développement d'un connecteur pour récupérer les flux de données depuis le Cloud UGA.
- Module de Parsing (Analyse syntaxique) : Implémentation de filtres (Regex ou analyseurs XML/JSON) pour isoler les variables critiques : identifiant unique de la décision, corps du texte de l'abstract et nomenclature de classification juridique.
- Module d'Insertion (ETL) : Script d'automatisation de l'injection SQL permettant d'assurer la cohérence des types de données et d'éviter les doublons (gestion des conflits d'index).

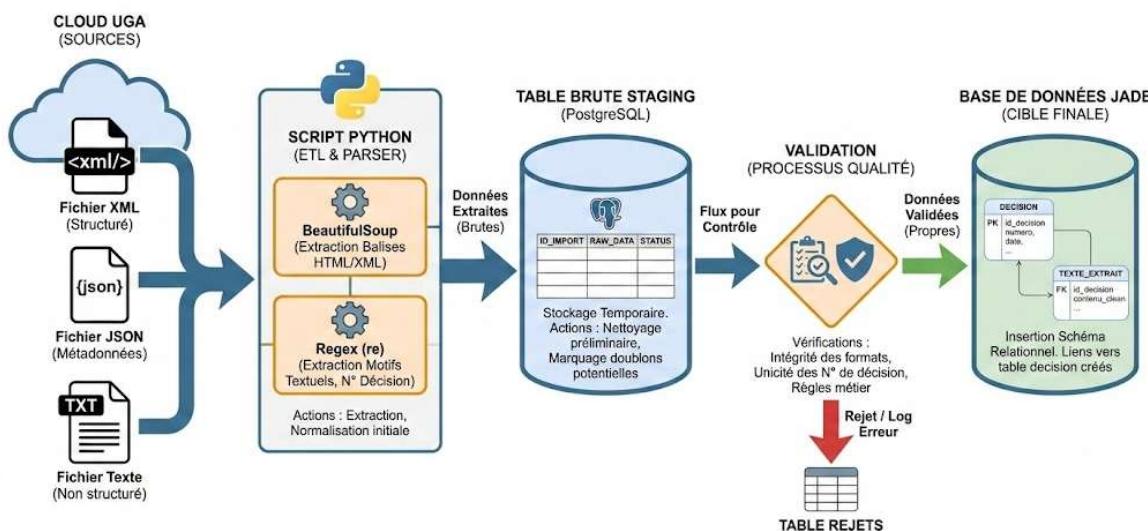


Figure 4 : Schéma fonctionnel du script d'intégration Python (Extraction - Transformation - Chargement).

## 4.c Contrôle de qualité et Fiabilisation

- Validation Métier (Juridique) : Test de conformité par échantillonage entre les données injectées et le site officiel du Conseil Constitutionnel (ex: référence 2013-4831 AN).
- Robustesse Technique : Mise en place d'un système de logs d'erreurs pour identifier et traiter les anomalies de formatage, garantissant qu'aucune analyse juridique ne soit ignorée par le script.

# 5. Ressources et contraintes

- Ressources Techniques :

- Environnement de développement : Python 3.x (gestion des dépendances via uv ou venv), PostgreSQL 13+.
  - Accès aux données : Fichiers sources (Abstracts 2023) hébergés sur le Cloud UGA et documentation du schéma existant.
  - Bibliothèques clés : SQLAlchemy (ORM), Pandas (Manipulation de données), Regex (Parsing).
- Contraintes du Projet :
- Intégrité des données (Critique) : Aucune perte d'information n'est tolérable lors du parsing ; le sens juridique de l'abstract doit être préservé à 100%.
  - Contrainte Temporelle : Le développement et les tests doivent tenir dans le volume horaire alloué (270h par étudiant), imposant une gestion stricte du planning.
  - Qualité logicielle : Utilisation impérative de Git (Gricad-GitLab) pour le versionnage et la traçabilité du code.

## 6. Planning et mise en œuvre (GANTT)

Le projet s'échelonne sur une période de 6 mois (Décembre 2025 – Juin 2026) et suit une méthodologie séquentielle divisée en quatre phases critiques :

Projet : JADE - Justice Algorithmique des Élections	
Date du début du projet :	08/12/2025
Date de fin du projet :	15/06/2026

Figure 5 : Diagramme de Gantt prévisionnel – Planification des phases de développement (Déc. 2025 – Juin 2026).

1. Phase 1 : Analyse & Conception (Décembre - Janvier)
  - Audit des fichiers sources et définition du modèle de données cible (MCD).
  - Jalon : Validation de la structure SQL "Abstracts".
2. Phase 2 : Développement (Février - Avril)
  - Développement itératif des scripts Python (Extraction, Regex, Insertion).
  - C'est la phase la plus dense en charge de travail technique.
3. Phase 3 : Fiabilisation (Mai)
  - Tests unitaires et vérification de la cohérence juridique des données intégrées.
4. Phase 4 : Livrables & Clôture (Juin)
  - Rédaction de la documentation technique, nettoyage du Git et soutenance.



Figure 6 : Priorisation des tâches dans le diagramme de Gantt

# 7. Organisation et répartition

Afin de respecter la contrainte pédagogique de 270 heures par étudiant tout en maximisant l'efficacité technique, les responsabilités ont été réparties par domaines de compétence, avec des phases de collaboration pour les tâches critiques.

## 7.a Rôles et Responsabilités

- RABEHI Milhane (Data Architect & SQL) :
  - Responsable de l'analyse de l'existant et de la modélisation relationnelle.
  - En charge de la création des tables et des requêtes SQL complexes.
  - Rédaction du cahier des charges et du rapport final.
- DJENNAD Mahdi (Backend Developer Python) :
  - Responsable de l'architecture du script d'automatisation.
  - Développement des modules de Parsing (Regex) et de nettoyage des données.
  - Gestion de la documentation technique du code (Git).

La collaboration est assurée par l'utilisation d'un dépôt Git pour le versionnage du code et par des points d'avancement hebdomadaires pour synchroniser les développements Python avec la structure de la base de données.

## 7.b Équilibre de la charge (270h)

Le tableau ci-dessous détaille la répartition horaire par tâche. Les phases d'analyse initiale et de tests finaux sont réalisées en binôme pour garantir la qualité, tandis que le développement est parallélisé.

TÂCHE	TEMPS EFFECTUÉ (H)/ MILHANE	TEMPS EFFECTUÉ (H)/ MAHDI	DÉBUT	FIN
Base de données relationnelle JADE	110	55	08/12/2025	20/01/2026
Analyse du schéma relationnel existant (UGA/IDEX)	15	15	08/12/2025	20/12/2025
Modélisation des tables pour les "Abstracts"	45	10	20/12/2025	05/01/2026
Création du script SQL (Tables/Index)	30	20	05/01/2026	15/01/2026
Optimisation de la base pour le texte	20	10	10/01/2026	20/01/2026
Automatisation Python & Parsing	85	145	21/01/2026	20/04/2026
Script de récupération des abstracts	10	30	21/01/2026	10/02/2026
Développement du parser (Regex/Text)	35	75	11/02/2026	25/03/2026
Script de nettoyage des données	15	15	26/03/2026	10/04/2026
Module d'insertion SQL automatique	25	25	11/04/2026	20/04/2026
Test de qualité des données	25	35	21/04/2026	20/05/2026
Tests d'intégrité (Double saisie/Script)	15	25	21/04/2026	10/05/2026
Debugging et robustesse du code	10	10	11/05/2026	20/05/2026
Livrables techniques	50	35	21/05/2026	15/06/2026
Documentation technique du code Git	10	20	21/05/2026	01/06/2026
Rapport final (Cahier des charges final)	35	10	01/06/2026	10/06/2026
Support de soutenance (Oral)	5	5	10/06/2026	15/06/2026
	270	270		

Figure 7 : Matrice de répartition de la charge de travail – Validation de l'équilibre des tâches (270h/étudiant)