L'IA PERO

Agent Intelligent Sémantique & Génératif

Adam BELOUCIF
Data Engineer & NLP

Amina MEDJDOUB
Frontend & Visualization

🎓 Projet Certifiant RNCP40875 • Bloc 2 • EFREI Paris • Février 2026

Résumé Exécutif

Objectif

Développer une solution IA capable de **comprendre des intentions complexes** (ex: "Je veux un cocktail réconfortant") là où les filtres classiques échouent.

Choix Techniques

- **Hybridation :** SBERT (Recherche) + Gemini (Génération).
- **Performance :** Latence < 50ms sur la recherche vectorielle.
- **Architecture :** RAG (Retrieval-Augmented Generation) avec format JSON strict.

Pipeline Hybride

Combinaison de la précision du **Data Engineering** et de la créativité de l'**IA Générative**.

Compétence Bloc 2 Validée

Imprimer / PDF

Problématique & Solution

Le Problème SQL

Les bases de données relationnelles ne comprennent pas le langage naturel ou les émotions.

```
Query: "Cocktail d'hiver réconfortant"
Result: 0 rows found.
```

La Solution Sémantique

Vectorisation des demandes pour trouver la proximité mathématique entre "Hiver" et "Whisky/Épices".

```
Vector Search: "Winter + Comfort"
Match: Old Fashioned (0.85)
```


Imprimer / PDF

Méthodologie & Organisation

Agile

Développement itératif.
Sprint 1: ETL & Data.
Sprint 2: NLP & RAG.
Sprint 3: UI Streamlit.

Binôme

Adam: Backend, NLP, API.
Amina: Frontend, UX, Tests.

Qualité Code

Utilisation de GitFlow.
Tests unitaires (Pytest).
Code documenté (Docstrings).

Imprimer / PDF

Pipeline de Données (ETL)

Collection

API & Scraping
Raw Data

Cleaning

Standardisation
Unités (ml/oz)

Enrichissement

Calcul Profils Goût
(Sucré/Amer...)

Embedding

SBERT
Vectorisation

Storage

CSV Clean +
Pickle Cache

Stack Technique : Python 3.11 Pandas Sentence-Transformers

Imprimer / PDF

Le Dataset Final

Métriques Clés

635

RECETTES UNIQUES

8

FEATURES / ITEM

Données nettoyées et enrichies manuellement pour garantir la qualité des résultats du RAG (Garbage In, Garbage Out).

DATA_SAMPLE.JSON


```
{ "id": "11007", "name": "Margarita", "ingredients":
["Tequila", "Triple Sec", "Lime"], "taste_profile": {
"Acidité": 4.5, "Alcool": 3.8, "Douceur": 2.0 },
"embedding": [-0.041, 0.123, ... 384 dims ...] }
```


Imprimer / PDF

Stratégie GenAI (RAG)

SYSTEM_PROMPT.PY

```
SYSTEM_PROMPT = """ ROLE: Expert Mixologue Années 20. TON:
Élégant, mystérieux, précis. INSTRUCTION: 1. Analyse la
demande : {user_query} 2. Utilise le contexte récupéré :
{rag_context} 3. Réponds UNIQUEMENT au format JSON strict.
"""
```

Architecture RAG

1. Retrieval (Recherche)
Identification des recettes similaires via Cosine Similarity (SBERT).

2. Augmentation
Injection des données fiables (Ingrédients, Dosages) dans le prompt.

3. Generation
Google Gemini génère la réponse finale en adoptant le persona.

Démonstration Live

Scénario : "Le Client Indécis"

Saisie
Requête Complexe

Visualisation
Radar Plotly

Génération
Recette IA

Imprimer / PDF

Résultats & Qualité

88%	24/24	1.8s	99%
PRÉCISION SÉMANTIQUE	TESTS PYTEST OK	TEMPS GÉNÉRATION	VALIDITÉ JSON

Validation Technique

Guardrail SBERT
Rejette efficacement les requêtes hors-sujet ("Réparer vélo").

Structure JSON
Le frontend Streamlit ne crashe jamais grâce au prompt strict.

Imprimer / PDF

Conclusion

Validation des Compétences RNCP

- [x] **C3.1 Préparation de données NLP & Structuration**
- [x] **C5.2 Développement solution IA Générative (RAG)**
- [x] **C5.3 Évaluation technique & Tests automatisés**

Adam BELOUCIF
Data Engineer

Amina MEDJDOUB
Frontend Dev

Imprimer / PDF