



# Manuel d'installation CoDIR IA

## Version Windows & macOS

---

**Dépôt officiel GitHub :** <https://github.com/RobinCop63/CoDIR-IA>

*(Disponible à partir du 1er décembre 2025)*

<b>CHAPITRE 1 – INTRODUCTION GÉNÉRALE</b>	<b>5</b>
1.1 Objectif du manuel	Erreur ! Signet non défini.
1.2 Public visé	5
1.3 Environnement de travail recommandé	5
1.4 Présentation du projet CoDIR IA	5
1.5 Structure du dossier type	6
<b>CHAPITRE 2 – PRÉ-REQUIS SYSTÈME</b>	<b>7</b>
2.1 Vérification de la version de Windows	7
2.2 Mise à jour du système	7
2.3 Installation de Python 3.13	7
2.4 Installation de Git	7
2.5 Vérification PowerShell 7+	7
2.6 Installation de Visual Studio Code	7
2.7 Vérification finale	8
<b>CHAPITRE 3 – INSTALLATION DE CODIR IA</b>	<b>9</b>
3.1 Création du dossier de travail	9
3.2 Ouverture du terminal PowerShell	9
3.3 Création de l'environnement virtuel	9
3.4 Activation de l'environnement	10
3.5 Installation des dépendances	10
<b>CHAPITRE 4 – CONFIGURATION INITIALE</b>	<b>11</b>
4.1 Le fichier.env	11
4.2 Vérification des chemins internes	11
4.3 Test de lecture du fichier.env	11

4.4 Configuration de Streamlit	11
4.5 Configuration réseau locale	12
<b>CHAPITRE 5 – LANCEMENT ET TEST</b>	<b>13</b>
5.1 Vérification de l'environnement	13
5.2 Lancement via le fichier.bat	13
5.3 Accès à l'interface	13
5.4 Tests de fonctionnement	13
5.5 Arrêt propre du serveur	13
5.6 Nettoyage et sauvegarde	14
<b>CHAPITRE 6 – RÉOLUTION DES PROBLÈMES COURANTS</b>	<b>15</b>
6.1 Le fichier launch.bat ne se lance pas	15
6.2 Erreur "Activate.ps1 ne peut pas être chargé"	15
6.3 Streamlit ne démarre pas	15
6.4 Clés API non reconnues	16
6.5 Python n'est pas reconnu	16
6.6 Conflit de port Streamlit	16
6.7 Interface figée ou lente	17
<b>CHAPITRE 7 – MISES À JOUR ET MAINTENANCE</b>	<b>18</b>
7.1 Mettre à jour les dépendances Python	18
7.2 Mettre à jour le code du projet (GitHub)	18
7.3 Sauvegarder régulièrement	18
7.4 Nettoyage de l'environnement	18
7.5 Contrôle des versions	18
7.6 Réinstallation complète	19

<b>CHAPITRE 8 – BONNES PRATIQUES ET SÉCURITÉ</b>	<b>20</b>
<b>8.1 Protéger les clés API</b>	<b>20</b>
<b>8.2 Confidentialité et données locales</b>	<b>20</b>
<b>8.3 Bonnes pratiques d'organisation</b>	<b>20</b>
<b>8.4 Sécurité Windows</b>	<b>20</b>
<b>8.5 Utilisation responsable</b>	<b>20</b>
<b>8.6 Prochaines étapes</b>	<b>20</b>
<b>ANNEXES A À D</b>	<b>22</b>
<b>Annexe A – Glossaire technique</b>	<b>22</b>
<b>Annexe B – Liste complète des bibliothèques Python</b>	<b>23</b>
<b>Annexe C – Historique des versions CoDIR IA</b>	<b>24</b>
<b>Annexe D – Crédits et remerciements</b>	<b>25</b>

# CHAPITRE 1 – INTRODUCTION GÉNÉRALE

## 1.1 Objectif du manuel

Ce manuel a pour objectif de guider pas à pas l'installation complète de CoDIR IA sur un environnement Windows 11 ou macOS (Intel / Apple Silicon).

Il s'adresse à toute personne souhaitant déployer localement le projet, même sans expérience préalable en développement Python.

L'objectif est d'obtenir un environnement fonctionnel, stable et sécurisé, capable d'exécuter le tableau de bord Streamlit de CoDIR IA.

## 1.2 Public visé

Ce guide est conçu pour:

- Les dirigeants, DAF, responsables de transformation ou consultants souhaitant tester CoDIR IA sur leur poste.
- Les utilisateurs disposant de notions de base en informatique.
- Les profils non techniques souhaitant installer une IA locale de manière simple et reproductible.

## 1.3 Environnement de travail recommandé

Composant	Recommandation minimale	Recommandation optimale
Système d'exploitation	Windows 10 (64 bits)	Windows 11 Pro 64 bits
Processeur	Intel i5 ou AMD équivalent	Intel i7 / Ryzen 7
Mémoire vive (RAM)	8 Go	16 Go ou plus
Stockage libre	10 Go minimum	50 Go sur SSD
Carte graphique	Facultative	NVIDIA RTX avec 8 Go VRAM
Connexion Internet	Oui, pour installation	Recommandée pour mises à jour
Logiciels	PowerShell 7+, VS Code, Python 3.13	idem

## 1.4 Présentation du projet CoDIR IA

CoDIR IA (pour Comité de Direction IA) est un orchestrateur multi-IA open source conçu pour accompagner les décideurs dans leurs prises de décision quotidiennes.

Il permet de connecter plusieurs modèles d'intelligence artificielle (OpenAI, Google Gemini, Anthropic, Mistral, etc.) dans une interface unique, simple et performante.

## 1.5 Structure du dossier type

Une fois installé, le dossier CoDIR IA ressemblera à ceci:

```
CoDIR-IA/
|
|— app.py
|— launch.bat
|— requirements.txt
|— .env
|— venv/
|  |— Scripts/
|  └─ Lib/
└─ providers/
   |— openai_provider.py
   |— gemini_provider.py
   └─ mistral_provider.py
```

## CHAPITRE 2 – PRÉ-REQUIS SYSTÈME

### 2.1 Vérification de la version de Windows

Avant toute installation, vérifie que ton système est à jour.

Ouvre le menu Démarrer → Paramètres → Système → Informations.

Vérifie que tu utilise Windows 10 ou 11 – 64 bits.

### 2.2 Mise à jour du système

1. Ouvre Paramètres → Windows Update.
2. Clique sur Rechercher des mises à jour.
3. Installe toutes les mises à jour disponibles.
4. Redémarre ton ordinateur.

### 2.3 Installation de Python 3.13

1. Rends-toi sur <https://www.python.org/downloads/windows/>
2. Télécharge Python 3.13 (Windows Installer 64-bit).
3. Coche Add Python to PATH avant d'installer.
4. Clique sur Install Now.
5. Ouvre PowerShell et vérifiez: `python --version`

### 2.4 Installation de Git

1. Vas sur <https://git-scm.com/downloads>
2. Télécharge la version Windows 64-bit.
3. Lance l'installation par défaut.
4. Vérifie dans PowerShell: `git --version`

### 2.5 Vérification PowerShell 7+

1. Ouvre PowerShell et tape: `$PSVersionTable.PSVersion`
2. Si la version est inférieure à 7.0, télécharge la dernière sur GitHub.

### 2.6 Installation de Visual Studio Code

1. Télécharge depuis <https://code.visualstudio.com/>
2. Installe avec les options par défaut.

3. Installe les extensions Python, Streamlit,.ENV files support, GitLens.

## 2.7 Vérification finale

Ouvre PowerShell et teste:

```
python --version
```

```
git --version
```

```
code --version
```



## CHAPITRE 3 – INSTALLATION DE CODIR IA

### 3.1 Création du dossier de travail

1 Ouvre l'Explorateur de fichiers.

2 Crée un dossier à la racine de ton disque ou dans OneDrive:

C:\Users\<ton\_nom>\OneDrive - Hi! Gestion\CoDIR-IA\

3 Copie dans ce dossier tous les fichiers du dossier public CoDIR IA (par exemple téléchargés depuis GitHub).

→ Il doit contenir au minimum:

app.py

launch.bat

requirements.txt

.env

providers/

outputs/

prompts/

💡 Évite les chemins trop longs ou contenant des espaces (cela cause parfois des erreurs Python).

### 3.2 Ouverture du terminal PowerShell

1 Clique dans la barre d'adresse du dossier et tape:

Powershell

2 Une fenêtre PowerShell s'ouvre directement dans ton dossier CoDIR IA.

Tu devrais voir:

```
PS C:\Users\<ton_nom>\<OneDrive - ... \CoDIR-IA> par exemple,
```

### 3.3 Création de l'environnement virtuel

L'environnement virtuel isole ton installation Python du reste du système.

Dans PowerShell, tape:

```
python -m venv venv
```

Une fois terminé, tu verras apparaître un dossier venv dans ton dossier CoDIR-IA.


### 3.4 Activation de l'environnement

Toujours dans PowerShell:

```
.\.venv\Scripts\activate
```

Tu sauras que tout fonctionne si ton invite de commande devient:

(venv) PS C:\Users\<ton\_nom>\<OneDrive - ...>\CoDIR-IA> par exemple,

 En cas d'erreur "Activate.ps1 ne peut pas être chargé", lance d'abord cette commande:

```
Set-ExecutionPolicy Unrestricted -Scope Process
```

### 3.5 Installation des dépendances

Installe maintenant tous les modules Python nécessaires:

```
pip install -r requirements.txt
```



Attends la fin de l'installation avant de continuer.  
Si tu vois des avertissements "already satisfied" ou "successfully installed", c'est bon signe.

## CHAPITRE 4 – CONFIGURATION INITIALE

### 4.1 Le fichier.env

Le fichier.env contient toutes les variables confidentielles du projet (clés API, ports, préférences). Tu peux l'ouvrir avec Visual Studio Code ou un simple Bloc-notes.

Exemple:

```
OPENAI_API_KEY=sk-xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
ANTHROPIC_API_KEY=sk-ant-xxxxxxxxxxxxxxxxxxxxx
GEMINI_API_KEY=AlzaSy...
MISTRAL_API_KEY=xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
PORT=8501
```

⚠ Ne partage jamais ce fichier sur Internet ou GitHub.

### 4.2 Vérification des chemins internes

Certains fichiers (comme launch.bat) contiennent une ligne qui pointe vers ton dossier local.

Ouvre launch.bat et vérifie la ligne suivante:

```
cd "C:\Users\<ton_nom>\OneDrive - Hi! Gestion\CoDIR-IA"
```

🧠 Le chemin entre guillemets doit correspondre exactement à ton dossier local CoDIR IA.

### 4.3 Test de lecture du fichier.env

Tu peux tester si les clés sont bien lues avec cette commande:

```
python -c "import os; print(os.getenv('OPENAI_API_KEY'))"
```

Si la clé s'affiche (ou au moins le début), c'est que ton fichier.env est bien détecté ✅

### 4.4 Configuration de Streamlit

Le tableau de bord CoDIR IA s'appuie sur Streamlit.

Tu peux vérifier sa bonne installation:

```
streamlit --version
```

Si la commande répond avec un numéro de version, tout est prêt.

Sinon:

```
pip install streamlit
```

## 4.5 Configuration réseau locale

Le port par défaut utilisé est le 8501.

Si tu veux changer (par exemple en cas de conflit réseau):

Ouvre le fichier.env

Modifie la ligne:

```
PORT=8502
```

Streamlit lancera alors l'application sur le nouveau port.

## CHAPITRE 5 – LANCEMENT ET TEST

### 5.1 Vérification de l'environnement

Assure-toi que ton environnement virtuel est actif:

(venv) PS C:\Users\<ton\_nom>\<OneDrive - ...>\CoDIR-IA> par exemple,

Si ce n'est pas le cas:

```
.\venv\Scripts\activate
```

### 5.2 Lancement via le fichier.bat

Double-clique sur launch.bat

→ Une fenêtre s'ouvre avec le message:

=====

 Lancement de CODIR IA


=====

Puis quelques secondes plus tard, Streamlit démarre ton tableau de bord.

### 5.3 Accès à l'interface

Ton navigateur s'ouvre automatiquement sur:

<http://localhost:8501>

 Si ce n'est pas le cas, ouvre-le toi-même et colle l'adresse.

### 5.4 Tests de fonctionnement

Une fois l'application affichée:

Vérifie que la page d'accueil s'ouvre sans erreur.

Teste chaque modèle IA (OpenAI, Gemini, etc.) pour valider les connexions.

Si tu obtiens une erreur "clé API manquante", retourne dans ton fichier.env.

### 5.5 Arrêt propre du serveur

Quand tu veux fermer CoDIR IA:

Reviens dans la fenêtre PowerShell.

Appuie sur CTRL + C pour interrompre Streamlit.

Puis ferme la fenêtre.

## 5.6 Nettoyage et sauvegarde

Avant de quitter définitivement:

Ferme ton environnement virtuel:

```
deactivate
```

Sauvegarde ton dossier CoDIR IA sur OneDrive.

Vérifie que le fichier.env est bien exclu de tout partage public.

## CHAPITRE 6 – RÉOLUTION DES PROBLÈMES COURANTS

### 6.1 Le fichier launch.bat ne se lance pas

Symptôme: rien ne se passe ou la fenêtre se ferme immédiatement.

Causes possibles:

Le chemin du dossier CoDIR IA dans le.bat est incorrect.

L'environnement virtuel venv n'a pas été créé.

Solution:

Ouvre ton fichier launch.bat avec le Bloc-notes.

Vérifie la ligne:

```
cd "C:\Users\<ton_nom>\OneDrive - Hi! Gestion\CoDIR-IA"
```

Si le chemin est correct, vérifie que le dossier venv existe.

Sinon, recrée-le:

```
python -m venv venv
```

### 6.2 Erreur “Activate.ps1 ne peut pas être chargé”

Cause: la politique d'exécution PowerShell bloque les scripts locaux.

Solution:

```
Set-ExecutionPolicy Unrestricted -Scope Process
```

Puis relance:

```
.\venv\Scripts\activate
```

### 6.3 Streamlit ne démarre pas

Symptôme:

ModuleNotFoundError: No module named 'streamlit'

Solution:

```
pip install streamlit
```

Si le problème persiste:

```
pip install --upgrade streamlit
```

## 6.4 Clés API non reconnues

Symptôme:

Erreur KeyError: 'OPENAI\_API\_KEY' ou message d'erreur dans Streamlit.

Solution:

Ouvre ton fichier.env.

Vérifie la syntaxe:

```
OPENAI_API_KEY=sk-xxxxxxxxxxxxxx
```

Redémarre CoDIR IA après modification:

```
deactivate  
.  
.\env\Scripts\activate  
streamlit run app.py
```

## 6.5 Python n'est pas reconnu

Symptôme:

```
python n'est pas reconnu en tant que commande interne
```

Solution:

Réinstalle Python en cochant "Add Python to PATH".

Ou ajoute manuellement ce chemin dans les variables d'environnement:

```
C:\Users\<nom>\AppData\Local\Programs\Python\Python313\
```

## 6.6 Conflit de port Streamlit

Symptôme:

OSError: [Errno 98] Address already in use

Solution:

Ouvre.env et change la ligne:

```
PORT=8502
```

Puis relance:

```
streamlit run app.py
```



## 6.7 Interface figée ou lente

Causes possibles:

Trop d'onglets ouverts dans le navigateur.

Le cache Streamlit est saturé.

Solution:

Ferme les autres onglets et vide le cache:

```
streamlit cache clear
```

## CHAPITRE 7 – MISES À JOUR ET MAINTENANCE

### 7.1 Mettre à jour les dépendances Python

Ouvre PowerShell dans ton dossier CoDIR IA.

Active l'environnement:

```
.\env\Scripts\activate
```

Mets à jour tous les modules:

```
pip install --upgrade -r requirements.txt
```

💡 Fais-le une fois par mois pour rester à jour.

### 7.2 Mettre à jour le code du projet (GitHub)

Si tu as cloné le projet via GitHub:

```
git pull origin main
```

Cela met à jour ton dossier local avec la dernière version publique de CoDIR IA.

### 7.3 Sauvegarder régulièrement

Crée une copie du dossier CoDIR-IA sur ton OneDrive ou ton NAS Synology.

À sauvegarder impérativement:

Le dossier /outputs

Tes prompts personnalisés

Le fichier.env

### 7.4 Nettoyage de l'environnement

Si tu veux repartir sur une base propre:

```
deactivate
```

```
rmdir /s /q venv
```

```
python -m venv venv
```

```
pip install -r requirements.txt
```

### 7.5 Contrôle des versions

Pour vérifier la version actuelle:

```
python --version  
streamlit --version  
git --version
```

Note-les dans un coin, utile pour les échanges de support.

## **7.6 Réinstallation complète**

Si tu veux tout réinitialiser:

Sauvegarde ton.env

Supprime tout le dossier CoDIR-IA

Télécharge la dernière version sur GitHub

Reprends les chapitres 3 à 5 du manuel

## CHAPITRE 8 – BONNES PRATIQUES ET SÉCURITÉ

### 8.1 Protéger les clés API

Ne partage jamais ton fichier.env.

Si tu envoies ton projet, crée une version publique sans clés.

Utilise des comptes API personnels, jamais partagés.

### 8.2 Confidentialité et données locales

CoDIR IA ne collecte ni n'envoie aucune donnée sans ton action.  
Les fichiers sont traités localement sur ton ordinateur.  
💡 Les logs générés par Streamlit peuvent être supprimés à tout moment.

### 8.3 Bonnes pratiques d'organisation

Regroupe tes fichiers de travail dans /outputs.

N'ajoute pas de fichiers volumineux dans /providers.

Garde une structure claire et identique entre machines (utile pour support).

### 8.4 Sécurité Windows

Active et maintiens à jour:

Windows Defender ou Bitdefender

Les mises à jour Windows automatiques

Les sauvegardes OneDrive

Un mot de passe fort sur ton compte Windows

### 8.5 Utilisation responsable

L'open source implique de la transparence et du respect:

Cite les sources quand tu réutilises le projet.

Participe à la communauté (retours, améliorations).

Ne diffuse jamais de contenus sensibles générés par les IA sans vérification humaine.

### 8.6 Prochaines étapes

Une fois cette version Windows maîtrisée, tu pourras:

Tester la version macOS (chapitres spécifiques à venir).

Contribuer à l'évolution du projet sur GitHub.

Rejoindre la communauté CoDIR IA (via LinkedIn ou le futur site).

## Annexes A à D

### Annexe A – Glossaire technique

Terme	Définition simplifiée
API Key	Clé d'accès unique fournie par un service (OpenAI, Google, etc.) pour authentifier les requêtes.
.env	Fichier contenant les variables d'environnement (clés, ports, etc.), jamais partagé publiquement.
Environnement virtuel (venv)	Espace Python isolé où sont installés les modules nécessaires au projet.
Streamlit	Framework Python permettant de créer des interfaces web interactives.
Git / GitHub	Outils de versionnage permettant de suivre les changements et collaborer sur un projet.
Prompt	Texte d'entrée envoyé à une IA pour générer une réponse.
Token	Unité de mesure d'un texte lu ou écrit par un modèle d'IA (~4 caractères anglais).
PATH	Variable système indiquant à Windows où trouver les programmes exécutables.
Requirements.txt	Liste des bibliothèques Python à installer pour le projet.
PowerShell	Terminal Windows moderne et puissant.
Streamlit App	Application web locale générée par Streamlit.
Orchestrateur multi-IA	Logiciel connectant plusieurs IA dans une interface unique.

## Annexe B – Liste complète des bibliothèques Python

Bibliothèque	Fonction principale
Streamlit	Interface utilisateur et tableau de bord
Openai	Connexion aux modèles GPT (ChatGPT, GPT-4, GPT-5)
anthropic	Connexion aux modèles Claude
google-generativeai	Connexion à Gemini (Google)
mistralai	Connexion aux modèles Mistral et Mixtral
python-dotenv	Lecture des variables du fichier .env
requests	Requêtes HTTP vers les API externes
pandas	Manipulation et affichage de données tabulaires
tiktoken	Calcul du nombre de tokens par message
langchain (optionnel)	Gestion de chaînes logiques d'agents IA
watchdog (optionnel)	Surveillance des fichiers pour rechargement auto
streamlit-option-menu	Interface améliorée pour menus et onglets
pyyaml	Lecture et écriture de fichiers YAML

## Annexe C – Historique des versions CoDIR IA

Version	Changements principaux
v14.0 (août 2025)	Première version stable, support multi-IA complet
v14.2	Correction erreurs Streamlit + mode test offline
v14.5	Intégration simplifiée du fichier .env et optimisation du lancement
v14.8	Compatibilité Gemini 2.0
v15.0 (octobre 2025)	Version publique open source (Windows)
v15.1 (prévision)	Version macOS + optimisation GPU NVIDIA/Apple Silicon



## **Annexe D – Crédits et remerciements**

### Développement et co-auteurs:

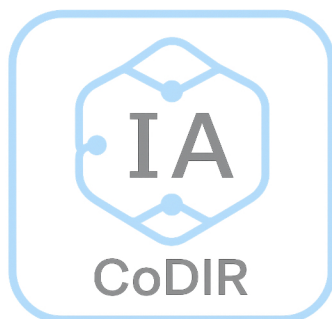
- Robin Sauzet – Créateur du projet CoDIR IA, conception, test et documentation
- ChatGPT (OpenAI GPT-5) – Co-auteur IA et générateur de documentation technique

### Contributions et frameworks:

- OpenAI – pour les modèles GPT-4 et GPT-5
- Anthropic – pour la série Claude
- Google – pour Gemini (via google-generativeai)
- Mistral AI – pour Mixtral
- Hugging Face – pour l'écosystème open source
- Streamlit – pour le framework du tableau de bord
- Python Software Foundation – pour la base du projet

Merci à tous ceux qui participent à rendre l'intelligence artificielle plus accessible, transparente et éthique. CoDIR IA est un projet collectif avant tout.

<b>Composant</b>	<b>Recommandation minimale</b>	<b>Recommandation optimale</b>
Système d'exploitation	Windows 10 (64 bits)	Windows 11 Pro 64 bits
Processeur	Intel i5 ou AMD équivalent	Intel i7 / Ryzen 7
Mémoire vive (RAM)	8 Go	16 Go ou plus
Stockage libre	10 Go minimum	50 Go SSD
Carte graphique	Facultative	NVIDIA RTX avec 8 Go VRAM
Connexion Internet	Oui, pour installation	Recommandée pour mises à jour
Logiciels	PowerShell 7+, VS Code, Python 3.13	idem



## **CoDIR IA – Manuel d’installation Windows**

**Version finale – Novembre 2025**

Auteurs : Robin Sauzet & ChatGPT (SASU Hi! Gestion)

Logo : Camille Belloir (<https://www.linkedin.com/in/camille-belloir/>)

Ne modifie jamais directement les fichiers du dépôt principal : crée une copie ou utilise une branche dédiée si besoin.

Sauvegarde régulièrement ton dossier local avant toute mise à jour.

### 3. Bonnes pratiques de versionning

Les principales évolutions sont également listées dans l'annexe C du manuel.

Consulte le fichier CHANGELOG.md ou l'historique des commits sur la page GitHub du projet.

### 2. Suivre l'historique des versions

Cette commande télécharge les dernières modifications du projet public.

```
git pull origin main
```

Utilise la commande suivante dans PowerShell, dans le dossier CoDIR-IA :

#### 1. Mettre à jour le code depuis GitHub

Ce chapitre explique comment maintenir à jour ton installation CoDIR IA grâce au versionning GitHub, et comment suivre les évolutions du projet.



## Mise à jour et suivi des versions GitHub

### ANNEXES A À D 15

#### ANNEXE A – GLOSSAIRE TECHNIQUE 15

#### ANNEXE B – LISTE COMPLÈTE DES BIBLIOTHÈQUES PYTHON 16

#### ANNEXE C – HISTORIQUE DES VERSIONS CODIR IA 17

#### ANNEXE D – CRÉDITS ET REMERCIEMENTS 17

## CHAPITRE 8 – Installation de CoDIR IA sur macOS (version courte)

### 8.1 Objectif de ce chapitre

Ce chapitre propose une version simplifiée de l'installation de CoDIR IA sur un Mac, pour des utilisateurs qui souhaitent principalement tester l'orchestrateur sans entrer dans tous les détails techniques. Pour une installation avancée ou du développement, il sera possible de créer plus tard un manuel dédié macOS.

### 8.2 Pré-requis côté macOS

Avant de commencer, vérifie les éléments suivants :

- Un Mac récent (idéalement macOS 13 Ventura ou version ultérieure).
- Une connexion Internet fonctionnelle (pour installer Python, Git et les dépendances).
- Python 3.10 ou supérieur installé (commande `python3` disponible dans le Terminal).
- Git installé (commande `git` disponible dans le Terminal).

Si Python 3 ou Git ne sont pas installés, tu peux les ajouter via Homebrew :

```
brew install python git
```

### 8.3 Récupération du dépôt CoDIR IA

1. Ouvre l'application Terminal (Spotlight → « Terminal »).
2. Choisis ou crée un dossier de travail (par exemple Documents/Projets).
3. Exécute les commandes suivantes :

```
git clone https://github.com/RobinCop63/CoDIR-IA.git
```

```
cd CoDIR-IA
```

### 8.4 Lancement simplifié via le script `launch.sh`

Le projet CoDIR IA inclut désormais un script de lancement compatible macOS (et Linux) nommé `launch.sh`.

1. Rends le script exécutable (à faire une seule fois) :

```
chmod +x launch.sh
```

2. Lance ensuite l'orchestrateur avec :

```
./launch.sh
```

Ce script va automatiquement :

- créer l'environnement virtuel venv/ si nécessaire ;
- installer les dépendances Python à partir de requirements.txt ;
- créer un fichier .env à partir de .env.template si aucun .env n'existe ;
- lancer l'interface utilisateur Streamlit dans ton navigateur par défaut.

## 8.5 Configuration du fichier .env sur macOS

Le fichier .env contient les clés API nécessaires pour appeler les différents modèles d'IA (OpenAI, Anthropic, Google Gemini, Mistral, etc.). Sans ces clés, CoDIR IA ne pourra pas interroger les modèles.

1. Ouvre le fichier .env créé à la racine du projet (copié depuis .env.template).
2. Renseigne chaque clé API que tu possèdes (OPENAI\_API\_KEY, ANTHROPIC\_API\_KEY, GOOGLE\_API\_KEY, MISTRAL\_API\_KEY, ...).
3. Tu peux laisser les paramètres de modèles par défaut, sauf si tu sais exactement ce que tu veux modifier.

Une fois les clés renseignées et le script launch.sh exécuté, ton Mac est prêt à exécuter une session CoDIR IA dans les mêmes conditions fonctionnelles que la version Windows.