

Exigences Fonctionnelles et Non Fonctionnelles

1. Exigences Fonctionnelles

1.1 Gestion des utilisateurs

- **EF1.1** : L'utilisateur doit pouvoir créer un compte sur la plateforme.
- **EF1.2** : L'utilisateur doit pouvoir s'authentifier (connexion / déconnexion).

1.2 Gestion des programmes

- **EF2.1** : L'utilisateur doit pouvoir sélectionner un programme préexistant parmi ceux proposés sur la plateforme.
- **EF2.2** : Le système doit permettre à l'utilisateur de paramétrer l'exécution (choix des données d'entrée, options de calcul).
- **EF2.3** : Le système doit permettre à l'utilisateur d'interrompre l'exécution.

1.3 Exécution sur le cluster

- **EF3.1** : Le système doit permettre de distribuer la charge de calcul entre les différents nœuds du cluster Raspberry Pi.
- **EF3.2** : Le système doit lancer les calculs en parallèle sur plusieurs nœuds.
- **EF3.3** : Le système doit collecter les résultats des différents nœuds.
- **EF3.4** : Le système doit afficher les résultats combinés à l'utilisateur sur l'interface web.

1.4 Interface web

- **EF4.1** : L'utilisateur doit pouvoir visualiser l'état du cluster (nœuds actifs, charge/performances, disponibilité).
- **EF4.2** : L'utilisateur doit pouvoir suivre l'exécution d'un programme en temps réel (historique).
- **EF4.3** : L'utilisateur doit pouvoir télécharger les résultats après exécution.

1.5 Historique et logs

- **EF5.1** : Le système doit enregistrer un historique des exécutions par utilisateur.
- **EF5.2** : L'utilisateur doit pouvoir consulter ses anciens résultats.

2. Exigences Non Fonctionnelles

2.1 Performance

- **ENF1** : Le cluster doit pouvoir exécuter des tâches distribuées efficacement, avec un temps de communication minimal entre les nœuds.
- **ENF2** : Le système doit supporter plusieurs utilisateurs simultanés sans dégradation significative des performances.
- **ENF3** : Le temps de réponse de l'interface web doit être minimisé pour les opérations standards.

2.2 Sécurité

- **ENF4** : Les mots de passe doivent être chiffrés en base de données.

2.3 Fiabilité

- **ENF5** : Le système doit être capable de détecter et gérer les pannes (message d'erreur) d'un ou plusieurs nœuds du cluster.
- **ENF6** : Les calculs doivent être rejouables ou récupérables en cas de défaillance partielle.

2.4 Maintenabilité

- **ENF7** : Le code source doit être documenté et versionné (Git/GitHub).
- **ENF8** : L'architecture doit être modulaire pour permettre l'ajout de nouvelles fonctionnalités (nouvelles tâches de calcul)

2.5 Accessibilité et ergonomie

- **ENF9** : L'interface doit être simple et intuitive, adaptée à des utilisateurs non experts.

2.6 Compatibilité et environnement

- **ENF10** : Le système doit fonctionner sur un cluster de 5 Raspberry Pi avec un OS compatible.