

GIT & GITHUB

4eme année IIR

BENTALBA

Salah eddine



Consultant expert
transformation digitale

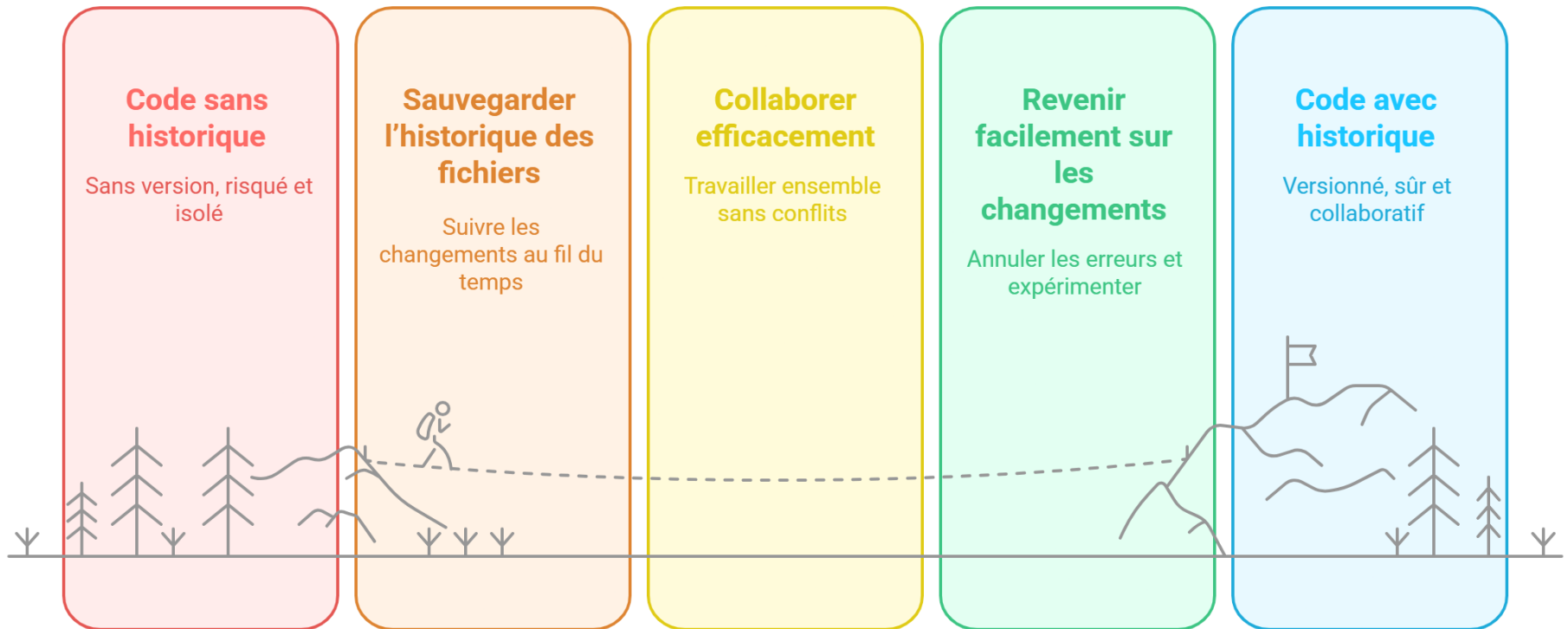
+212 601 100 600

Salaheddine.bentalba@gmail.com

- Lauréat EMSI Casablanca 2012
- Co-Fondateur de **Mobiconnect Technologies**
- **13+ ans d'expérience** en gestion de projets IT et transformation digitale.
- Spécialiste des **services cloud**.
- Collaborations avec **ATOS**, **Capgemini**, **42Gears** et **SOTI**.
- Interventions lors de **conférences et formations** :
 - Coach & Expert dans **EMSI NextGen Hackathon** et d'autres.
 - Organisateur de plusieurs **TEDx** et conférencier.

Pourquoi utiliser GIT ?

Git : Machine à remonter le temps pour le code



Concepts clés GIT



Branches

Développement parallèle et isolé de fonctionnalités



Commit

Enregistrement de l'état du projet avec un message



Zone de préparation

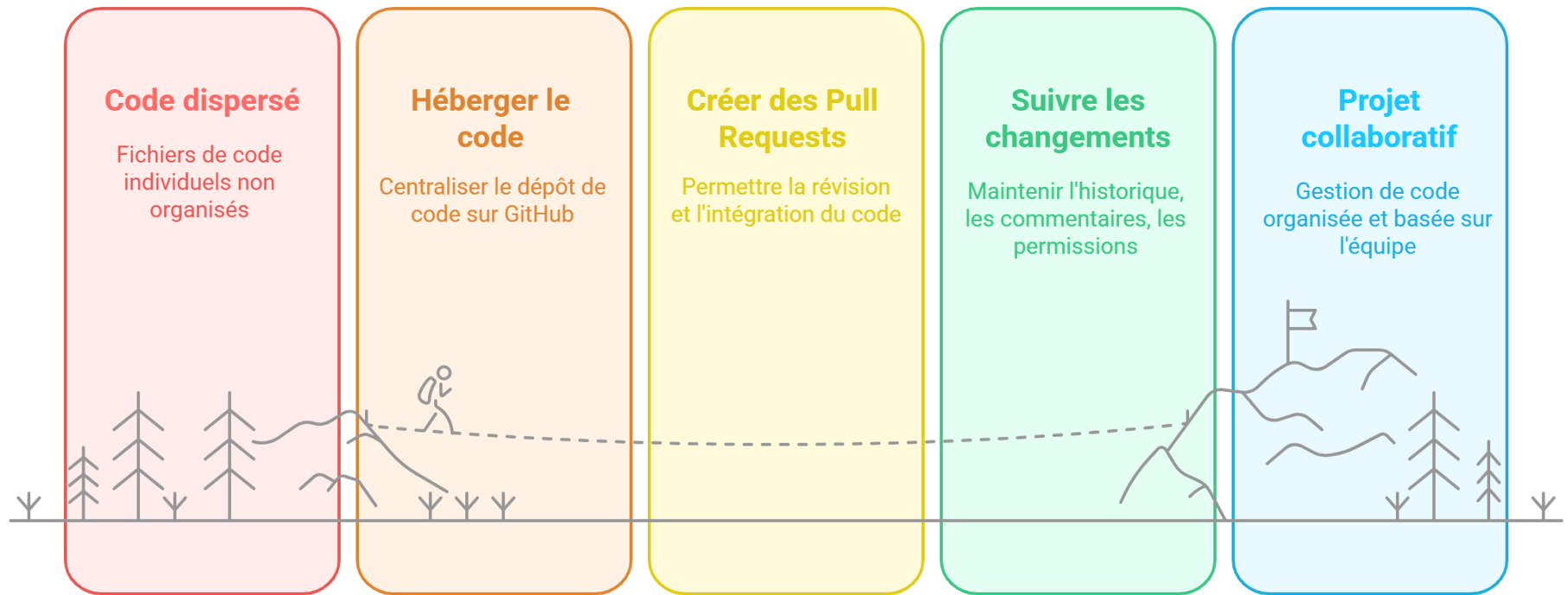
Zone temporaire pour les changements sélectionnés



Dépôt

Espace de stockage du code versionné

Pourquoi utiliser GITHUB ?



Comment collaborer avec GITHUB ?

Resolve Conflicts

Résoudre les différences avant la fusion.

Fork Repository

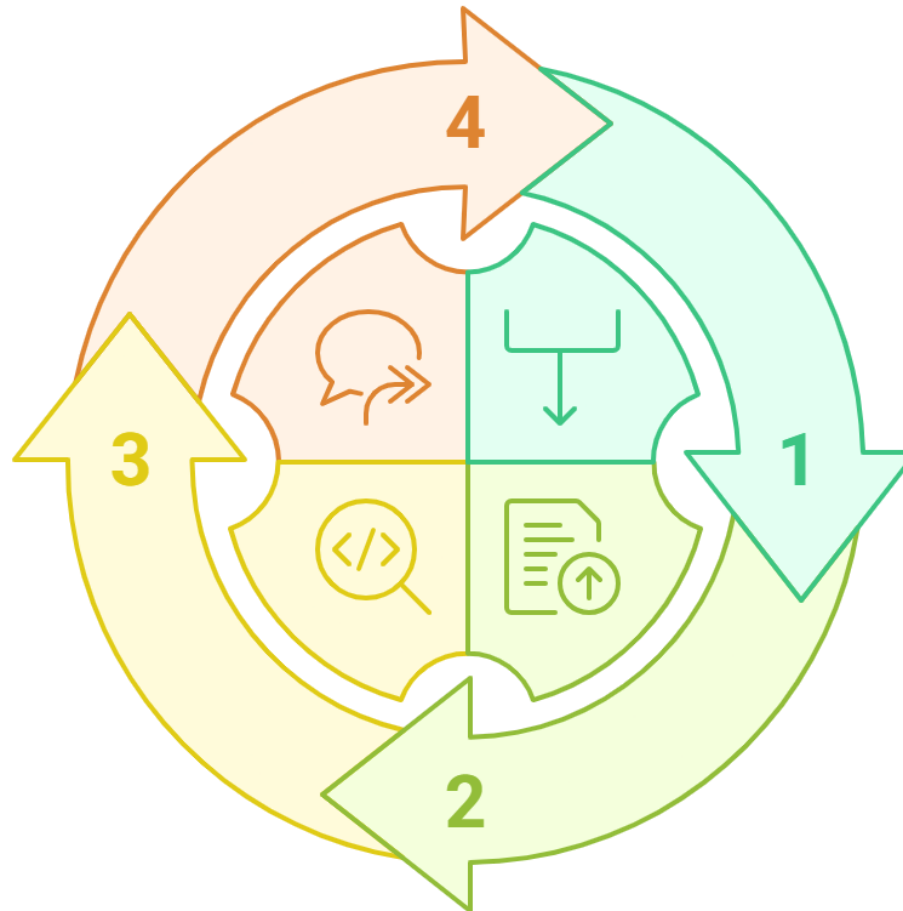
Créer une copie du dépôt pour les modifications.

Submit Pull Request

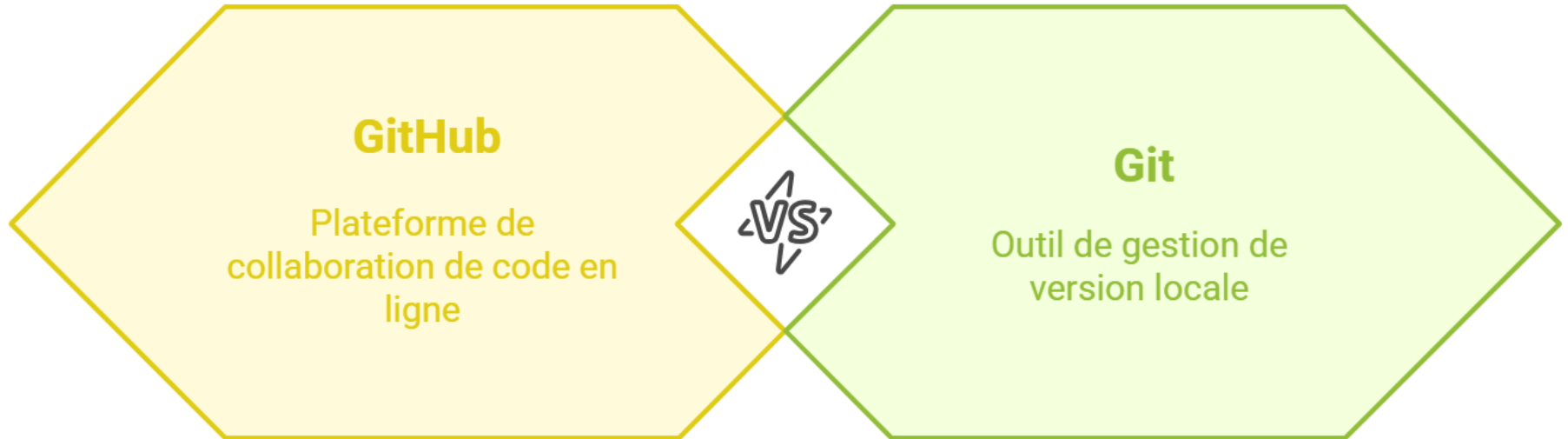
Proposer des changements pour révision.

Conduct Code Review

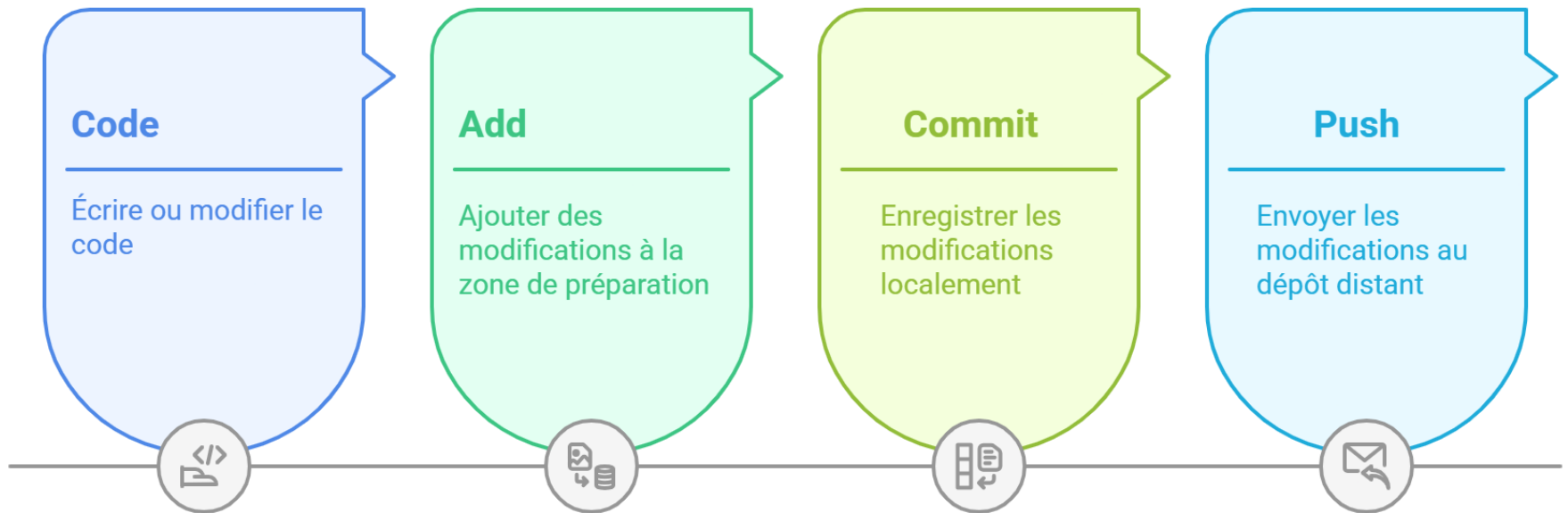
Examiner et commenter les changements proposés.



Git, GitHub... quelle différence ?



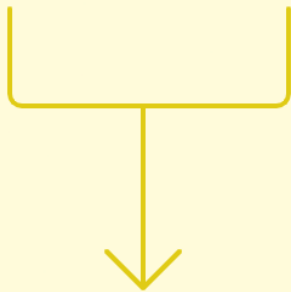
Workflow GIT



Branches de GIT – Travailler sans casser

Définition d'une branche

Une branche est une copie du projet pour tester ou modifier.



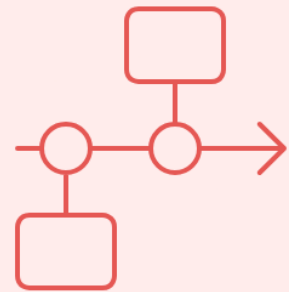
Prévention des ruptures

Les branches empêchent de casser le code principal pendant le développement.



Exemples de branches

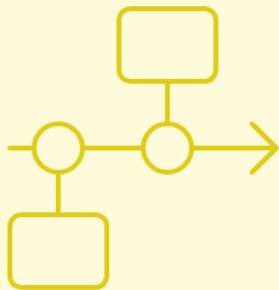
Les branches d'exemple incluent `main` et `feature-contact`.



Branches de GIT – commandes de base

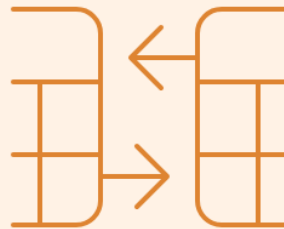
git branch

Liste, création et suppression de branches dans Git.



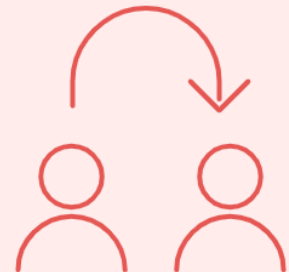
git checkout

Change de branche pour travailler sur différentes versions.



git merge

Fusionne les modifications d'une branche dans une autre.



Bonnes pratiques



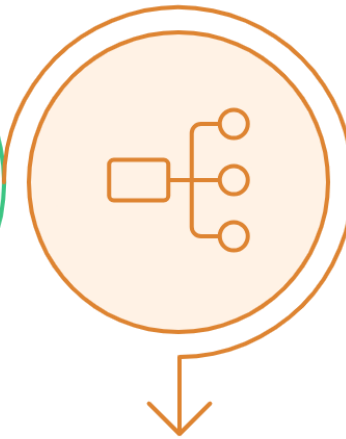
Messages clairs

Écrire des messages de validation précis et concis.



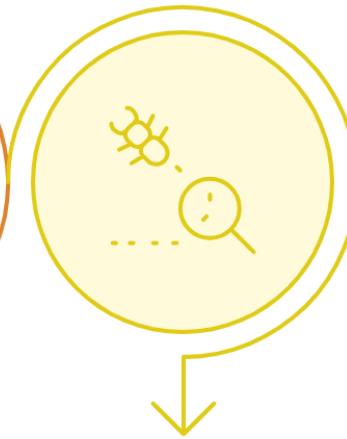
Commits fréquents

Valider régulièrement pour faciliter le suivi.



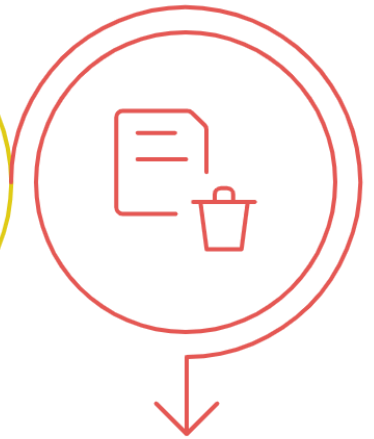
Branches dédiées

Utiliser des branches pour chaque nouvelle fonctionnalité.



Tests unitaires

Assurer la qualité du code via des tests automatisés.



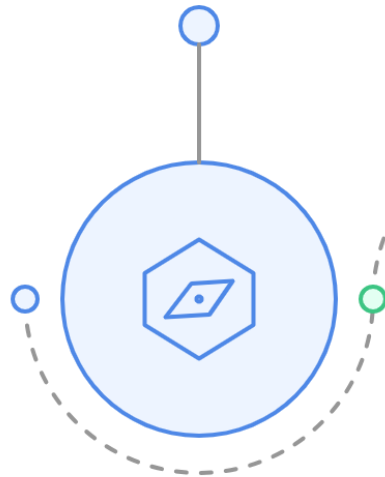
Dépôt propre

Maintenir un projet organisé et sans fichiers inutiles.

Ce que vous allez faire

Initialiser un projet Git

Créer un nouveau dépôt Git



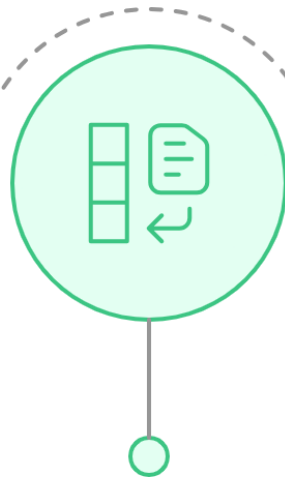
Travailler avec des branches

Gérer différentes lignes de développement



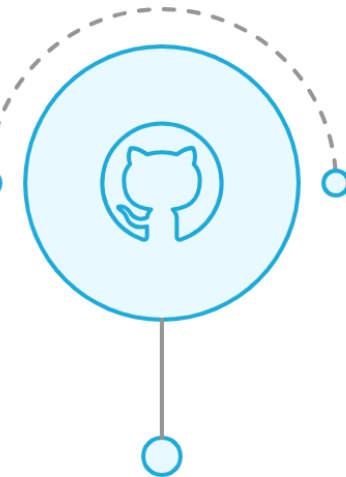
Versionner vos fichiers

Suivre les changements dans les fichiers



Pousser et collaborer sur GitHub

Partager et travailler ensemble sur GitHub



MERCI

Salaheddine.bentalba@gmail.com

+212 601 100 600