**Les bonnes pratiques du développement**

1. **Git et Jira**
2. Nommage des branches

* **feature**: Ajout d’une nouvelle fonctionnalité
* **bugfix**: Correction d’un bug
* **hotfix**: Correction d’un bug critique
* **chore**: Nettoyage du code
* **experiment**: Expérimentation de fonctionnalités

Pour créer une nouvelle branche pour une fonctionnalité : feature/nom-de-la-fonctionnalité

bugfix/*nom-du-bug/numero-du-tiket* pour un bug

hotfix/*nom-du-bug/numero-du-tiket*  pour un bug critique

1. Nommage des messages de commits

Le message du doit être précis et clair de sorte à comprendre facilement les modifications apportées au code.

Il doit contenir :

* Le type du commit :
* **feat** pour ajouter une nouvelle fonctionnalité
* **fix** pour la correction d’un bug
* **build** pour un changement lié aux dépendances
* **refactor** si le changement est lié au refactoring
* **test** : le ajout de nouveaux test unitaires
* **revert** : annulation de commit
* Le scope : la partie de l’architecture impactée (Controller, donnée etc.)
* Le sujet : courte description du commit
* Le message du commit : détail du commit
* Le footer : référence au ticket Jira

Pour nous assurer de l’application de ces règles de commit, nous installerons un hook de git.

1. Développement

**Notation des variables et des fonctions**

Utilisation du **lower camel case** : la première lettre de la chaine en minuscule et la première lettre de chaque mot de la chaine en majuscule

Exemple : prixViande pour indiquer le prix de la viande

**Les constantes**

La notation **Screaming snake case** est à utiliser pour nommer les constantes.

Cela consiste à mettre tous les mots de la constante en majuscule separés par les tirets du bas (\_).

Exemple : PRIX\_VIANDE .

**Commentaire**

Les fonctions, variable et les classes doivent être décrites de façon concise doit être brièvement