

# Tests dans le Projet TFDefectGA

---

Le document suivant présente les tests unitaires et d'intégration qui sont implémentées pour ce projet.

## Tests Unitaires

Les tests unitaires dans TFDefectGA se concentrent sur la vérification du comportement isolé des composants individuels :

### Catégories de tests unitaires

#### 1. Tests des parsers

- `test_terraform_parser.py` : Contient des tests qui assurent que le parseur Terraform peut analyser correctement différents types de structures dans les fichiers Terraform, gérer les cas limites, et fournir un comportement cohérent face aux entrées invalides.

#### 2. Tests des extracteurs de métriques

- `test_code_metrics_extractor.py` : Contient un test s'assure que l'extracteur de métriques peut traiter correctement les blocs Terraform, appeler l'outil d'analyse externe (terrametrics) et retourner les résultats dans le format attendu, tout en gérant proprement les ressources temporaires.
- `test_delta_metrics_extractor.py` : Contient des tests garantissent que l'extracteur de métriques delta peut calculer avec précision les changements dans les métriques de code entre différentes versions, tout en gérant correctement les cas d'erreur potentiels.
- `test_process_metrics.py` : Contient des tests garantissent que le module de calcul des métriques de processus fonctionne correctement et peut extraire des informations pertinentes sur l'historique de développement des blocs Terraform, ce qui est essentiel pour la prédiction des défauts.

#### 3. Tests des cas d'utilisation

- `test_analyze_tf_code.py` : Contient un test qui s'assure que la classe AnalyzeTFCode délègue correctement l'extraction des métriques à l'extracteur approprié et retourne les résultats sans les modifier, ce qui est crucial pour la fiabilité de l'analyse du code Terraform dans le système de prédiction de défauts.
- `test_detect_tf_changes.py` : Contient un test qui s'assure que la classe DetectTFChanges interagit correctement avec la classe GitChanges pour récupérer les blocs Terraform modifiés dans le dépôt, ce qui est essentiel pour le suivi des changements dans le code Terraform.
- `test_contribution_builder.py` : Contient des tests pour les fonctions qui construisent les informations de contribution pour les blocs Terraform. Ces tests vérifient que le système peut correctement extraire et organiser les données historiques des contributions aux fichiers Terraform.
- `test_feature_vector_builder.py` : Contient un test qui vérifie que le FeatureVectorBuilder peut correctement intégrer des métriques provenant de sources diverses en un format utilisable par le modèle de prédiction.

#### 4. Tests des factories

- `test_metrics_extractor_factory.py` : Contient un test qui vérifie que le pattern Factory est correctement implémenté pour les extracteurs de métriques, permettant au système de créer le bon type d'extracteur selon le contexte d'utilisation, tout en maintenant un couplage faible entre les composants.

## Tests d'Intégration

Les tests d'intégration dans TFDefectGA vérifient le fonctionnement coordonné de plusieurs composants :

#### 1. Test du processus complet de prédiction

- `test_integration_prediction.py` : Simule une exécution complète du workflow de prédiction, depuis l'extraction des métriques jusqu'à la mise à jour de l'historique des défauts

#### 2. Test de la construction des vecteurs de caractéristiques

- `test_integration_vector_builder.py` : Vérifie que le `FeatureVectorBuilder` fusionne correctement les différentes métriques pour alimenter le modèle prédictif

#### 3. Test d'Intégration du Modèle Random Forest

- `test_integration_randomforest_model.py` : Vérifie le fonctionnement du modèle `RandomForestModel`, qui est l'un des modèles de prédiction de défauts utilisés dans le système

## Exécution des Tests

Le projet dispose de plusieurs commandes pour exécuter les tests :

```
# Exécuter tous les tests
pytest tests/

# Exécuter uniquement les tests unitaires
pytest tests/unit/

# Exécuter uniquement les tests d'intégration
pytest tests/integration/

# Exécuter un fichier de test spécifique
pytest tests/unit/test_terraform_parser.py
```

Ces tests constituent un élément crucial de l'assurance qualité du projet TFDefectGA, garantissant que les différentes parties du système fonctionnent correctement de manière individuelle et collective.