

# Cahier de tests – Méthodologie TDD (Test-Driven Development)

## Objectif

Valider que chaque fonctionnalité développée respecte les attentes fonctionnelles, tout en adoptant une approche **TDD** : écrire les tests **avant** le code métier.

## Stratégie de tests

Type de test	Outils
Tests unitaires	JUnit 5 (backend), Vitest/Jest (front)
Tests intégration	Spring Boot Test (MockMvc)
Tests API	Postman + Newman ou REST Assured
Tests end-to-end	Cypress (web), Detox (mobile)

## Structure des tests backend (Spring Boot)

### Exemple : création d'un produit

@Test

```
void testCreerProduit_valide() {  
    Produit produit = new Produit("Farine", 10.0, "kg", 2.5,  
        LocalDate.now(), LocalDate.now().plusMonths(1));  
    Produit saved = produitRepository.save(produit);  
    assertNotNull(saved.getId());  
    assertEquals("Farine", saved.getNom());  
}
```

## Exemple : menu dépassant le seuil

@Test

```
void testCoutMenuDepasseSeuil() {  
    Menu menu = new Menu("Repas Luxueux", ...);  
    menu.setCoutTotal(5.5); // seuil max = 4.5 €  
    assertTrue(menu.getCoutTotal() > seuilBudget);  
}
```

## Jeux d'essai (données de test)

Scénario	Entrée	Attendu
Produit avec date de péremption	Farine - péremption demain	Alerte affichée
Menu économique	Menu < 2 €	Aucun avertissement
Stock négatif	Utilisation > stock disponible	Erreur bloquante
Produit expiré	Date < aujourd'hui	Inutilisable dans menu
Coût moyen > seuil	Menu à 5€, seuil à 4€	Alerte budget déclenchée

## Tests d'intégration (Spring Boot)

Utilisation de @SpringBootTest, @MockMvc et @Testcontainers :

@Test

```
void integrationTestAjoutProduitViaApi() throws Exception {  
    mockMvc.perform(post("/api/produits")  
        .contentType(MediaType.APPLICATION_JSON)  
        .content("{ \"nom\": \"Pâtes\", \"quantite\": 20 }"))  
        .andExpect(status().isCreated());  
}
```

# Tests Frontend (React)

## Exemple avec Vitest (React 18 + Tailwind)

```
test('affiche une alerte si le produit est périmé', () => {  
  render(<ProduitItem produit={{ nom: "Lait", datePeremption: "2024-  
01-01" }} />);  
  expect(screen.getByText(/périmé/i)).toBeInTheDocument();  
});
```