ATDD y Gherkin: tutorial práctico

Este tutorial introduce lo necesario para trabajar con **Acceptance Testing-Driven Development (ATDD)** usando **Gherkin** en proyectos como este repositorio.

El objetivo no es cubrir BDD avanzado ni escribir escenarios ultra detallados de UI, sino aprender a:

- Entender una especificación ATDD aunque no tengas más documentación adicional
- Escribir escenarios de aceptación claros
- Convertirlos en tests ejecutables
- Usarlos para guiar implementación y validación

¿Qué es ATDD?

ATDD es una forma de desarrollo guiada por **criterios de aceptación**.

En lugar de empezar por código, se empieza por ejemplos de comportamiento esperable del sistema.

- Se acuerdan ejemplos entre negocio/producto, desarrollo y testing
- Esos ejemplos se expresan en lenguaje entendible (Gherkin)
- Se automatizan como tests de aceptación
- Se implementa el sistema hasta que esos tests pasan

ATDD vs TDD vs BDD (visión corta)

- **TDD**: guía el diseño del código a nivel técnico (tests unitarios primero)
- **ATDD**: guía el desarrollo desde requisitos de negocio verificables

¿Qué aporta Gherkin en ATDD?

Gherkin es un formato de texto estructurado que permite describir comportamientos de forma legible y automatizable.

Estructura base:

```
Feature: Nombre de la capacidad del sistema

  Scenario: Comportamiento esperado
    Given contexto inicial
    When ocurre una acción
    Then se observa un resultado
```

Palabras clave principales:

- `Feature` : capacidad o bloque funcional
- `Scenario` : ejemplo concreto de comportamiento
- `Given` : estado inicial o precondiciones

Nivel de detalle recomendado para ATDD

Para ATDD, escribe escenarios que validen valor funcional, no scripts de clics.

Buen nivel de detalle:

- Reglas de negocio y resultados observables
- Datos importantes para entender el comportamiento
- Condiciones de éxito y error relevantes

Evitar (si no es imprescindible):

- Secuencias de UI de muy bajo nivel ("click en botón X", "abre modal Y")
- Detalles técnicos internos (clases, tablas, endpoints internos)
- Escenarios enormes que mezclan muchos comportamientos

Mini catálogo de requisitos ATDD (incluido en este tutorial)

Para practicar sin depender de otros ficheros, aquí tienes un ejemplo completo en un dominio habitual: **tienda online**.

Requisitos funcionales (ejemplo)

```
Feature: Compra de productos con stock limitado

  Background:
    Given existe el producto "Auriculares Bluetooth" con precio 59.99 EUR
    And el producto tiene 3 unidades en stock

  Scenario: Compra confirmada con pago aprobado
    Given la clienta "Lucia" ha añadido 1 unidad al carrito
    When confirma la compra con una tarjeta válida
    Then el pedido queda en estado "confirmado"
    And el stock disponible pasa a 2
    And se envía un email de confirmación

  Scenario: Compra rechazada por falta de stock
```

Flujo de trabajo ATDD (paso a paso)

1. Entender un requisito de negocio
2. Escribir o refinar su escenario Gherkin
3. Ejecutar tests de aceptación (fallarán al principio)
4. Implementar lo mínimo para hacerlos pasar
5. Refactorizar y repetir

Este ciclo puede convivir con TDD unitario.

Ejemplo práctico mínimo

1) Crear un fichero `.feature`

Ubicación habitual en proyectos Java con Cucumber:

`src/test/resources/features/checkout.feature`

Contenido:

```
Feature: Checkout de pedidos

  Background:
    Given existe el producto "Teclado mecánico" con precio 89.90 EUR
    And el producto tiene 2 unidades en stock

  Scenario: Compra confirmada con pago aprobado
    Given la clienta "Marta" está autenticada
    And tiene 1 unidad del producto en su carrito
    When confirma el pago con tarjeta válida
    Then el pedido queda confirmado
```

Plantilla recomendada para escenarios ATDD

```
Feature: <capacidad de negocio>

  Scenario: <comportamiento esperado>
    Given <contexto relevante>
    When <acción o evento principal>
    Then <resultado observable>
    And <resultado adicional>
```

Reglas prácticas:

- Un escenario = un comportamiento principal
- Nombres orientados a negocio
- `Then` verificable de forma objetiva
- Datos concretos cuando aporten claridad

Scenario Outline (sólo cuando compensa)

Cuando el mismo comportamiento se repite con varios datos, usa `Scenario Outline`.

```
Scenario Outline: Validación de acceso según rol
  Given un usuario con rol <rol>
  When intenta acceder al panel de administración
  Then el acceso es <resultado>

Examples:
  | rol         | resultado |
  | admin       | concedido |
  | cliente     | denegado  |
  | invitado    | denegado  |
```

Úsalo para evitar duplicación, no para ocultar lógica compleja.

Requisitos no funcionales en Gherkin

También se pueden expresar en formato ATDD:

```
Scenario: Tiempo máximo de checkout bajo carga
  Given existen 300 usuarios concurrentes comprando
  When una clienta confirma su pedido
  Then el tiempo total de respuesta es menor a 1500 ms
```

Claves:

- Métricas concretas (`< 2 segundos`, `HTTP 401`, etc.)
- Condiciones reproducibles
- Resultado medible

Checklist de calidad para escenarios ATDD

Antes de dar por bueno un escenario:

- [ ] ¿Describe un comportamiento de negocio relevante?
- [ ] ¿Es entendible por personas no técnicas?
- [ ] ¿El resultado (`Then`) es observable y medible?
- [ ] ¿Evita detalles de implementación innecesarios?
- [ ] ¿Puede automatizarse?
- [ ] ¿Está alineado con el requisito del hito?

Errores frecuentes

- Escribir escenarios como pruebas unitarias disfrazadas
- Mezclar demasiadas reglas en un único `Scenario`
- Usar pasos ambiguos ("funciona correctamente")
- No definir claramente precondiciones en `Given`
- Describir sólo camino feliz y olvidar errores relevantes

Tarea práctica propuesta

1. Elige uno de los escenarios del mini catálogo incluido en este tutorial
2. Crea un fichero `.feature` con ese escenario (o refínalo)
3. Añade al menos un escenario alternativo de error
4. Ejecuta tests con `mvn test`
5. Implementa lo mínimo para que pasen
6. Documenta en el PR qué requisito queda cubierto

Referencias

- Gherkin reference: https://cucumber.io/docs/gherkin/reference/
- Cucumber docs: https://cucumber.io/docs/