

# Trabajo Práctico 1: BDD

## 75.48 Calidad en desarrollo de software FIUBA

Dentro del proyecto Repartir se debe agregar una regla y dos o más escenarios que la ilustren, y luego implementar la regla en el código del proyecto (<https://github.com/dfontde/repartir-2024>). La regla a agregar es libre, aunque no debe estar ya implementada en el proyecto. Un ejemplo que pueden utilizar es “El total de un grupo no debe ser negativo”.

### Condiciones de entrega

- Hacer un fork del proyecto en GitHub.
- Los escenarios deben cumplir con las características BRIEF
  - Orientado al **negocio**/dominion (B de Business)
  - Datos **reales** cuando sea pertinentes (R de Real data)
  - Revelar **intención** (I de Intention revealing)
  - **Esencial**, sin nada que no haga falta para ilustrar el comportamiento
  - **Enfocado** ilustran una única regla (F de focused)
- Se debe escribir primero la regla, luego los escenarios, luego implementar las pruebas de aceptación automatizadas y finalmente la solución, siguiendo el ciclo de BDD.
- Las pruebas de aceptación automatizadas pueden ser del tipo *acceptanceTest* (punta a punta con Selenium) o *fastAcceptanceTest* (sobre los objetos de dominio). Se recomienda este último tipo de prueba para quienes hayan tenido problemas de entorno (por ejemplo, Docker o WebDriver).
- Se debe entregar link al repositorio público de GitHub con el código del proyecto pasando todas las pruebas (`./gradlew check`), e incluyendo la nueva funcionalidad implementada (la excepción la constituyen pruebas que no puedan correr por problemas de entorno como se describe en el ítem anterior).

Se recomienda probar manualmente la solución cuando las pruebas automatizadas estén pasando.

Para solucionar el tp, se implemento de la siguiente manera

## 1-Definicion de la regla a implementar

"El total de un grupo no debe ser negativo".

## 2-Escenarios (utilizando los Criterios BRIEF)

Usando los principios BRIEF (Orientado al negocio, Datos reales, Revelar intención, Esencial y Enfocado), definimos los siguientes escenarios en formato Gherkin:

**Se agrego en esta ruta: src/test/resources/features/total\_grupo\_no\_negativo.feature**

El archivo total\_grupo\_no\_negativo.feature que consta del siguiente texto

```
# language: es
Característica: Validación del total del grupo
  Como administrador de un grupo de gastos
  Quiero asegurar que el saldo total no sea negativo
  Para mantener la integridad contable del grupo
```

```
Escenario: Un grupo nuevo comienza con total en cero
  Dado que creo un grupo llamado "Viaje a Bariloche"
  Entonces el total del grupo debe ser 0 pesos
```

```
Escenario: No se puede registrar un gasto que resulte en un total negativo
  Dado que el grupo "Asado" tiene un total de 1000 pesos
  Cuando intento registrar un gasto de -1500 pesos por "Devolución"
  Entonces se debe rechazar la operación por "Saldo negativo no permitido"
  Y el total del grupo "Asado" debe permanecer en 1000 pesos
```

---

## 3-implementacion de las pruebas rapidas de aceptacion (sin modificar los originales)

En la ruta main\src\fastAcceptanceTest\java\ar\com\grupoesfera\repartir\steps  
Se agrego el archivo GrupoStepDefinitions.java

## 4-implementacion de la solucion en el dominio

Se implemento en la carpeta principal la clase grupo.java