AS

TDD

- **ITERATION 1:** Refactoring
    - Afegir a l'opearació de la Implementation la nova referència a una strategy concreta (default). Els tests no passaran (no està definit ni l'objecte de tipus Strategy ni l'operació). L'IDE ens ajuda a crear els d'Strategy necessaris i la interface.
    - Crear la classe default que implementi la Strategy i moure el codi anterior a la nova classe. Els test seguiran sense funcionar.
    - Refactoritzar l'operació setUp del test case per tal de que en crear l'objecte "x" s'usi l'Strategy definida (default). L'IDE ens ajuda a crear la constructora de l'objecte *x*". S'haurà de definir el codi de la constructora (amb paràmetre de Strategy). Ara els test han de passar. El codi fa el mateix que abans, però utilitzant el patró Strategy.

- **ITERATION 2:** Definir un nou test case, amb els testos relacionats amb el mètode d'aquesta estràtegia.
    - Definir correctament l'operació setUp() per tal de que en crear l'objecte *x* s'usi aquesta nova estratègia. L'IDE ens guia per crear la classe de la nova estratègia que implementa la interface Strategy.
    - Definir el test per provar l'estratègia.
    - Definir el codi de la operació per tal de que el test anterior passi.
    - Tantes iteracions com casos d'ús hi hagi en aquesta estratègia.

- **ITERATION 3:** Definir un nou test case, amb els testos relacionats al mètode d'aquesta estràtegia.
    - Definir correctament l'operació setUp() per tal de que en crear l'objecte *x* s'usi aquesta nova estratègia. L'IDE ens guia per crear la classe de la nova estratègia que implementa la interface Strategy.
    - Definir el test per provar l'estratègia.
    - Definir el codi per tal que el test anterior passi.
    - Tantes iteracions com casos d'ús hi hagi en aquesta estratègia.

- **ITERATION 4:**
    - Definir un test case TestIntegration amb *n* testos per provar que les *n* estrategies funcionen correctament.