**Trabajo Práctico N⁰ 2**

**Técnicas de Diseño**

**Grupo 10**

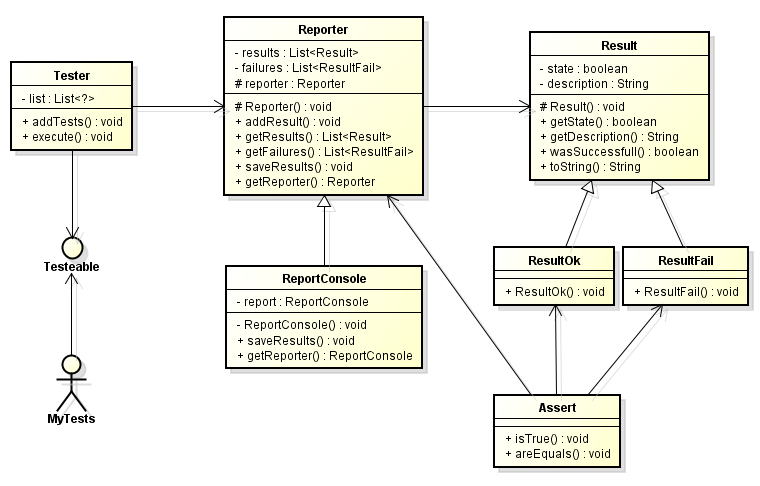
**Integrantes:**

81052 – Cesar Leguizamon ([fabi1816@gmail.com](mailto:david.mdq89@gmail.com))

90110 – David Marcos ([david.mdq89@gmail.com](mailto:david.mdq89@gmail.com))

91299 – Diego Meller ([mellerster@gmail.com](mailto:mellerster@gmail.com))

**Diagrama de clases**



**Clases e Interfaces, con sus responsabilidades**

Clase Result

Esta clase tiene el estado del resultado, exitoso o fallido, y una descripción del mismo.

Clase ResultOk

Esta clase se utiliza para separar los casos de resultados exitosos de los fallidos, en este caso es el exitoso y lo que hace es heredar de Result y cuando se crea le dice que es un resultado exitoso y a la descripcion se le agrega el texto para que cuando se grabe el reporte se notifique eso.

Clase ResultFail

Esta clase se utiliza para separar los casos de resultados exitosos de los fallidos, en este caso es el fallido y lo que hace es heredar de Result y cuando se crea le dice que es un resultado fallido y a la descripcion se le agrega el texto para que cuando se grabe el reporte se notifique eso.

Clase Reporter

Esta clase se encarga de ir agregando los resultados para generar un reporte.

Clase ReportConsole

Esta clase tiene como responsabilidad, enviar los resultados de las pruebas a la consola y un comentario final si pasaron o no todos los tests.

Clase Assert

Esta clase tiene como responsabilidad hacer las pruebas sobre los datos que se envían y agregar los resultados (correctos o incorrectos) al reporte.

Clase Tester

Su responsabilidad es agregar los testeables para poder ejecutar los test del cliente uno a uno y guardar los resultados del reporte.

Interfaz Testeable

Es lo que implementa el cliente para ejecutar una serie de tests, debe hacer que sus tests sean ejecutados dentro del método init().

**Herencias**

Clase ResultOk

Esta clase hereda de Result ya que simplifica para separar las pruebas exitosas de las fallidas y hacer los reportes de una manera más sencilla.

Clase ResultFail

Esta clase hereda de Result ya que simplifica para separar las pruebas exitosas de las fallidas y hacer los reportes de una manera más sencilla.

Clase ReportConsole

Esta clase hereda de Reporter ya que es un reporte, pero en este caso al grabarse (metodo saveResults()) lo muestra por pantalla.

**Comentarios adicionales**

Al ejecutar el programa se puede observar lo siguiente:

[Ok] --> ReportCountIncreaseTest: Expected true, value true

[Ok] --> ReportCountIncreaseTest: 1 and 1 are equals

[Ok] --> ReportFailCountIncreaseTest: 1 and 1 are equals

[Ok] --> ReportFailCountNotIncreaseTest: Expected true, value true

[Ok] --> ReportFailCountNotIncreaseTest: 1 and 1 are equals

[Ok] --> ResultOkStateTest: true and true are equals

[Ok] --> ResultFailStateTest: false and false are equals

[Ok] --> AssertIsTrueSuccessfullResultTest: Expected true, value true

[Ok] --> AssertIsTrueSuccessfullResultTest: true and true are equals

[Ok] --> AssertIsTrueNotSuccessfullResultTest: false and false are equals

[Ok] --> AssertEqualsSuccessfullResultTest: 1 and 1 are equals

[Ok] --> AssertEqualsSuccessfullResultTest: true and true are equals

[Ok] --> AssertEqualsNotSuccessfullResultTest: false and false are equals

[Fail] --> ReportFailCountIncreaseTest: Expected true, value false

[Fail] --> AssertIsTrueNotSuccessfullResultTest: Expected true, value false

[Fail] --> AssertEqualsNotSuccessfullResultTest: 1 and 2 are not equals

Test Fail: 3 Tests Failed

En realidad los 3 tests que marca como fallidos, fueron realizados a propósito para testear los resultados fallidos, los cuales cada uno de ellos luego funciona correctamente.