**Trabajo Práctico N⁰ 2.1**

**Técnicas de Diseño**

**Grupo 10**

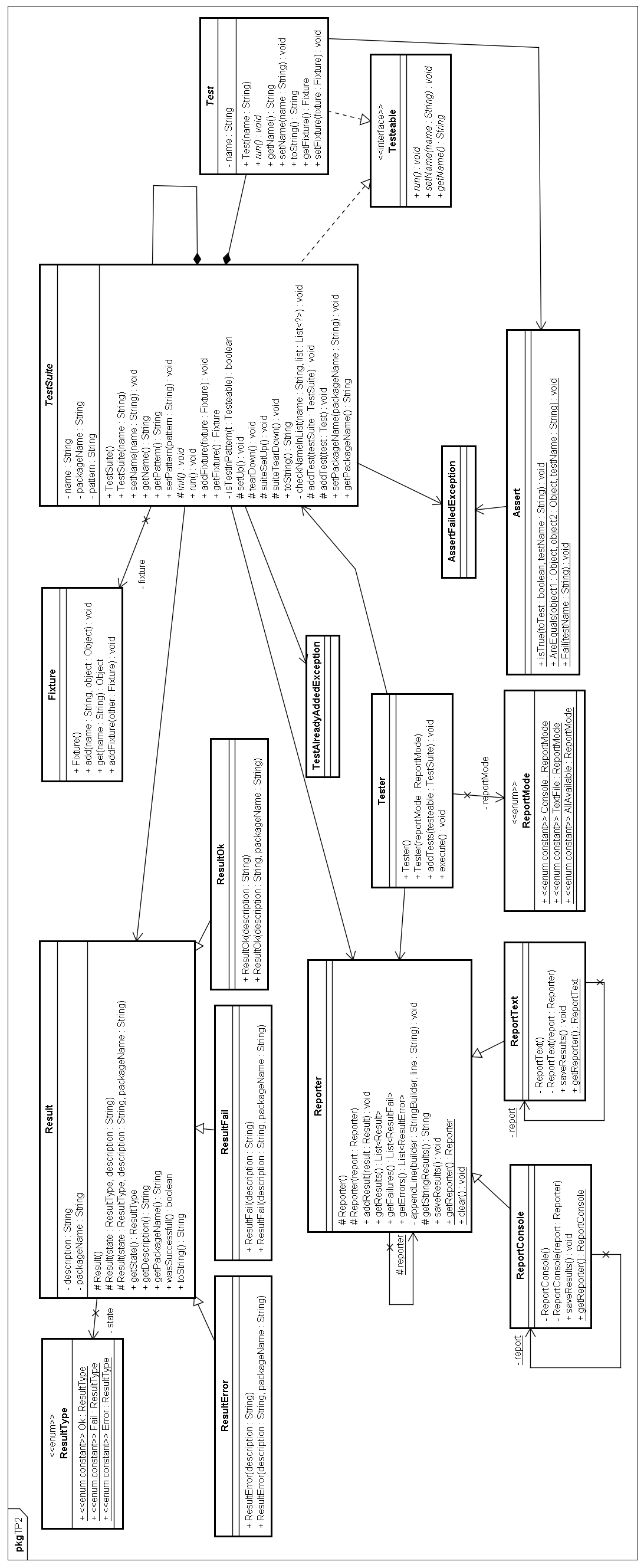
**Integrantes:**

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**

Reporter

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

ReportConsole

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

ReportText

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

ReportMode

Enumerado con los distintos tipos de reportes que hay: por consola, en archivo de texto, o todos los disponibles.

Assert

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

AssertFailedException

Exception que se lanza cuando no se devuelve el resultado que se esperaba en un assert.

Result

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

ResultType

Enumerado con los distintos tipos de resultados que hay: ok, fallidos o con errores.

ResultOk

Esta clase se utiliza para separar los casos de resultados exitosos de los fallidos y con errores. 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 vea eso.

ResultFail

Esta clase se utiliza para separar los casos de resultados exitosos de los fallidos y con errores. 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 descripción se le agrega el texto para que cuando se grabe el reporte se vea eso.

ResultError

Esta clase se utiliza para separar los casos de resultados exitosos de los fallidos y con errores. En este caso es con errores y lo que hace es heredar de Result y cuando se crea le dice que es un resultado con errores y a la descripción se le agrega el texto para que cuando se grabe el reporte se vea eso.

Testeable

Es una interfaz que define los métodos que deben implementar las clases que sean testeables.

TestSuite

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

Test

Esta clase representa los test individuales que se desean probar.

Tester

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

TestAlreadyAddedException

Exception que se lanza cuando se crea un test con el mismo nombre que otro creado anteriormente.

Fixture

Clase que representa la lista de objetos que son instanciados antes de correr los tests.

**Herencias**

ReportConsole

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

ReportText

Esta clase hereda de Reporter ya que es un reporte, pero en este caso al grabarse (metodo saveResults()) lo graba en un archivo de texto.

ResultOk

Esta clase hereda de Result ya que simplifica para separar las pruebas

exitosas de las fallidas y con errores, y hacer los reportes de una manera más sencilla.

ResultFail

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

ResultError

Esta clase hereda de Result ya que simplifica para separar las pruebas

exitosas de las fallidas y con errores, y hacer los reportes de una manera más sencilla.