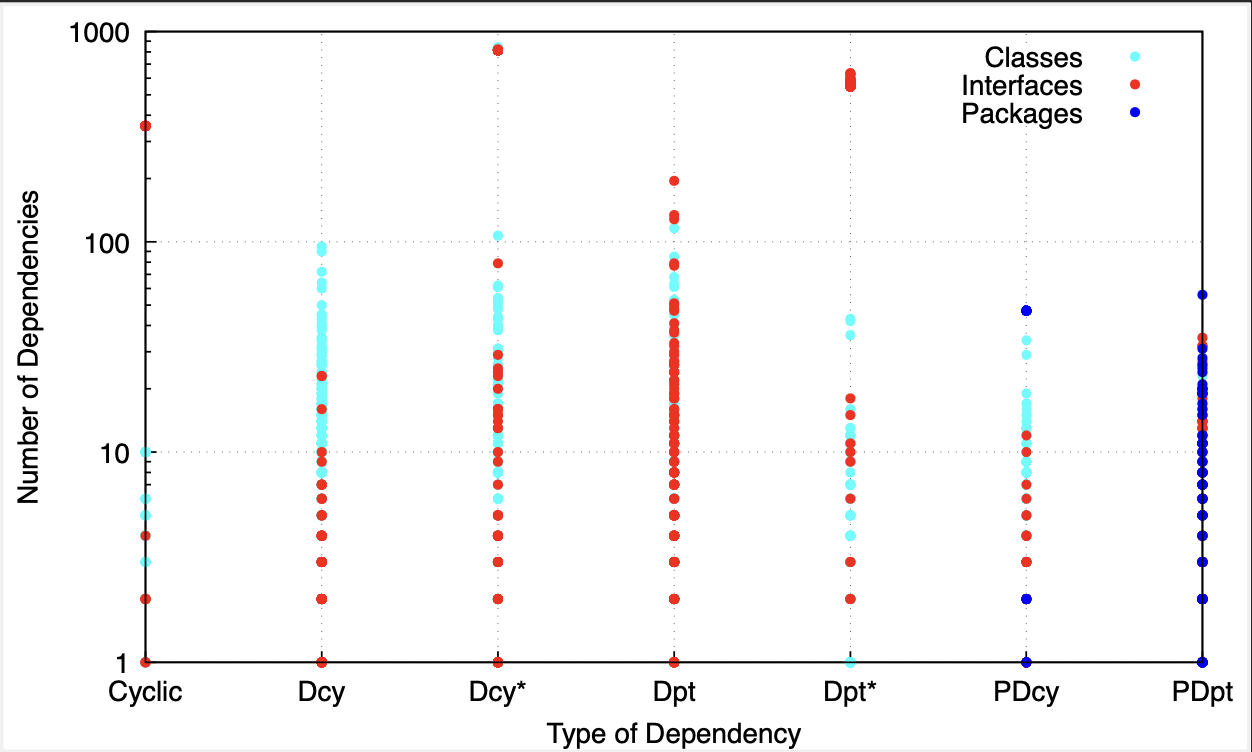
**Dependency Metrics** - Francisco Vale nº 60201



Cyclic: Calculates the number of classes or interfaces which each class directly or indirectly depends on, and which in turn directly or indirectly depend on it. Such cyclic dependencies may result in code which is difficult to understand and test.

Dcy: Calculates the number of classes or interfaces which each class directly depends on.

Dcy\*: Calculates the number of classes or interfaces which each class directly or indirectly depends on.

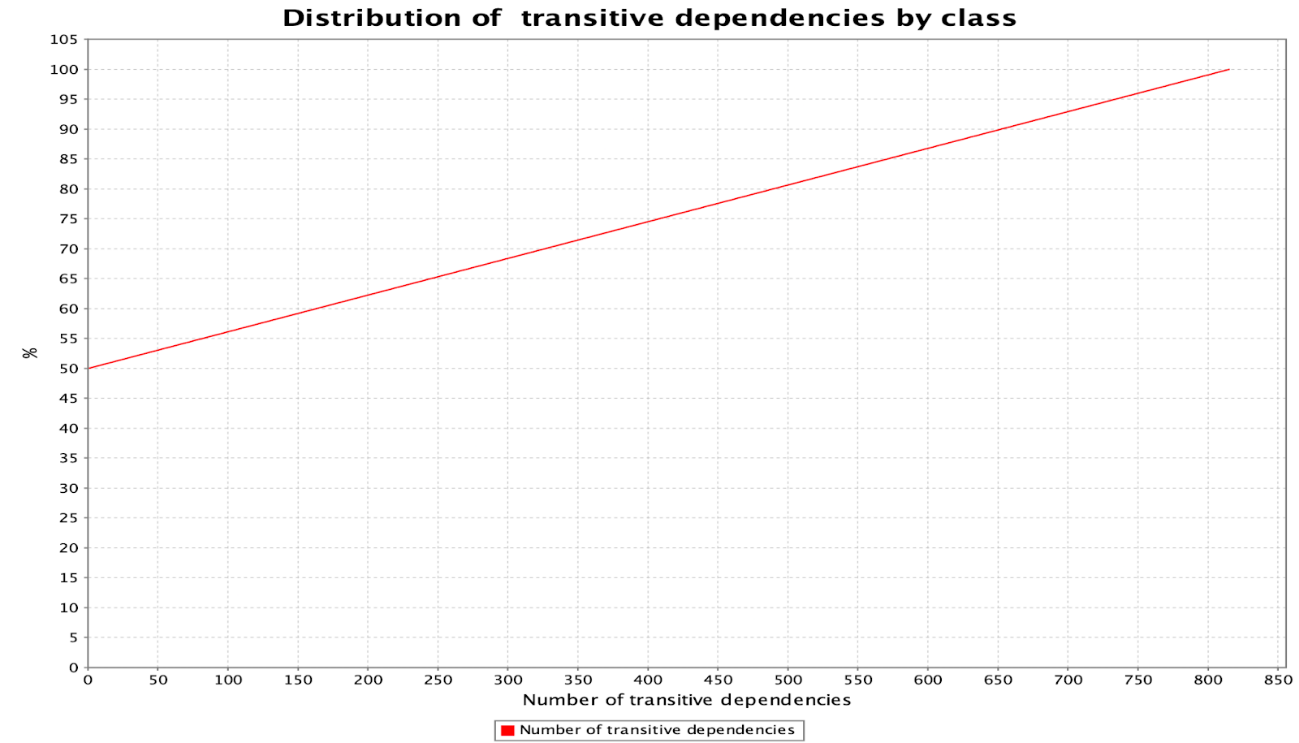
Dpt: Calculates the number of classes or interfaces which directly depend on each class.

Dpt\*: Calculates the number of classes or interfaces which directly or indirectly depend on each class.

PDcy: Calculates the number of packages on which each class directly or indirectly depends.

PDpt: Calculates the number of packages which directly or indirectly depend on each class.

As we can see there are some exceptional high values in the number of interfaces which each class directly or indirectly depends on. Searching for those classes I found the one which stood out, the Task Interface.



This Interface has around 800 classes dependent on it, directly or indirectly and approximately 50% of every class has some kind of dependency. This might bring out some code smells and so, if we intended to refactor our code and to test it this should be one of the Interfaces to check first.