Reporting tool

## Things to report

* Writing to an XML file to give some sort of intelligence to the tool
* Ability to merge different log messages to a common type message
* Treat different entries according to their types
* Use of JaroWinkler class from uk.ac.shef.wit.simmetrics.similaritymetrics
* Destroy every single structure after running
* Use of scanner instead of bufferedreader

## Objective

The purpose of this is for letting us build a proper tool useful to get reports on errors thrown by OGI’s applications during their runtimes. Since most of the “red ones” are ignored during development, we want to keep track of them all.

With this document, we want to get to know every single use that we can and want to take from this app.

## User Stories

As a user, I want to get reports from every single application and eBroker site in one single report.

As a user, I want to filter the severity of the errors to be showed on the reports based on a list of all the severe types that are available.

As a user, I want to configure the application so I can schedule it to give me reports of any OGI’s application or eBroker site’s runtime logs n-times a week, at a certain time and save it.

As a user, I want to edit or delete any scheduled report at any point.

As a user, I want to abort a currently running report.

As a developer, I want to make the system intelligent enough to add a new severe/info severity type to the lists as soon the parser finds a new one.

## Performance

## Add new applications

To add new applications to the reporter, you’ll have to go to:

* Create a new class inside lib.gui.applications.specific package named as **\*AppName\*ApplicationPanel.java** which extends ApplicationPanel.java (follow the existing ones to match consistency)
* **ApplicationsPanel.java**: add it in the *private static final String \*appName\*Application* section and then add it a new switch case inside addNewApplicationPanel using the previously created class.
* **OpenReportConfig.java:** add it wherever you find appropriate (for now it is in an alphabetical order)
* **ReportConfig.java:** and add it inside the init() method as **this.applications.put(ApplicationsPanel.\*appName\*Application, null);**

## Temp

**-------PORTUGUESE-------**

Falta implementar algumas coisas. Nomeadamente:

* Leitura do XML para popular os campos da secção “*Severity*”, para o filtro
* Fazer o grafismo
  + Configurador
  + E-mail enviado ao utilizador
* Classe responsável pelo envio dos e-mails
* Fazer o calendarizador
* Multithreading
* Verificar as classes LogData
* Importantissimo: ainda existem caminhos para ficheiros hardoced.
* Support dev’s ainda não tem lista definida
* Parsear os elementos passados no campo “Others”
* API-first design
  + Criar criadores de coisas, acho eu.