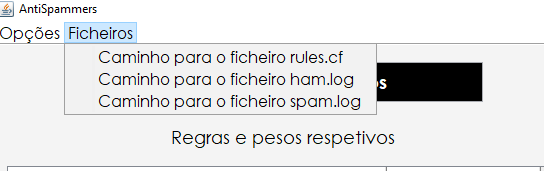
**SPRINT 1**

* **Estudo dos requisitos do Programa**

Esboço da GUI:



* Caminhos dos ficheiros
* Botão Aleatório
* Botão NSGA II
* Falsos positivos
* Edição dos pesos
* Gráfico
* Menu
* Informação
* Guardar no ficheiro
* **Caminhos dos ficheiros rules.cf, ham.log, spam.log**

Permite ao utilizador definir os caminhos dos ficheiros rules.cf, ham.log, spam.log, utilizando uma GUI.

* **Leitura dos ficheiros rules.cf, ham.log e spam.log**

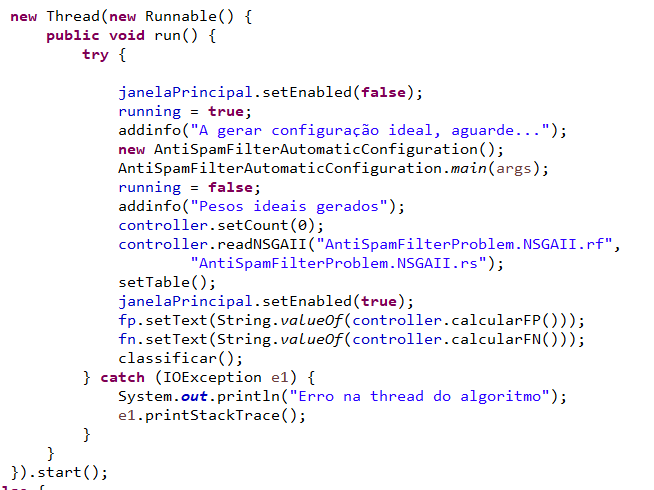
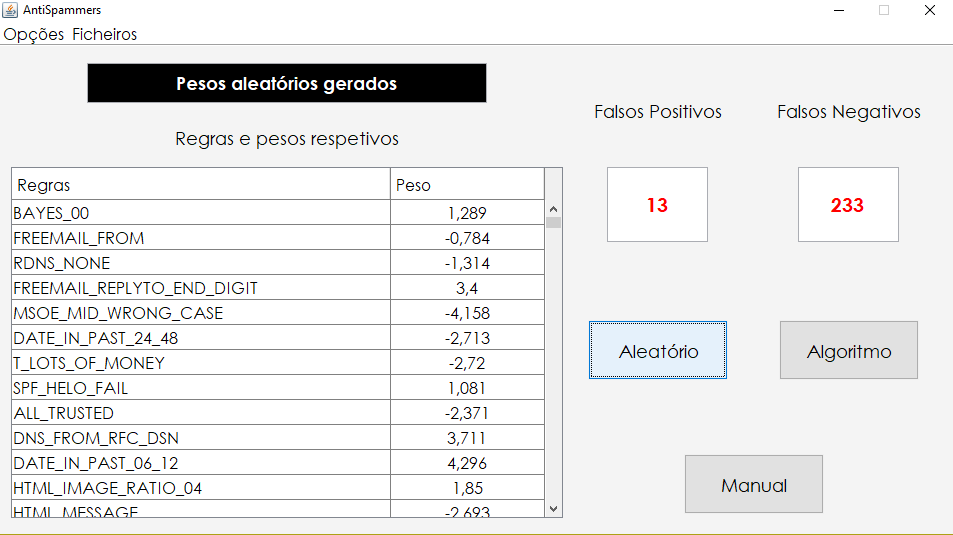
Após termos as Opções dos caminhos dos ficheiros temos que poder ativar a funcionalidade de ler os ficheiros, de forma a tornar o botão que selecionarmos executável.

**Sprint Retrospective 1:**

Este é o primeiro Sprint Retrospective há muitas funcionalidades ainda por implementar, mas já conseguimos criar um esboço da GUI que queremos, e avançamos também a janela idealizada no tal esboço, tendo conseguido configurar as janelas de leitura dos ficheiros rules.cf, ham.log e spam.log, e desenhado todas as outras ainda sem implementações.

No próximo Sprint esperamos ter mais janelas com funcionalidades implementadas.

**SPRINT 2**

* **Thread autónoma para algoritmo**
* **Apresentar os resultados de FP (falsos positivos) e FN (falsos negativos) para a respetiva configuração**
* **Desenvolvimento de código de cores como classificação da configuração atual**
* **Permitir ao utilizador gerar uma configuração aleatória**
* **Calculo dos FP e FN**

**Sprint Retrospective 2:**

Neste Sprint avançamos, como tínhamos planeado no Sprint 1, a estrutura da GUI. Colocámos em funcionamento as varias opções de cálculo, criando uma thread autónoma para algoritmo.

Para o próximo Sprint responsabilizamo-nos em tentar finalizar as funcionalidades das restantes opções implementadas na janela da GUI tentando chegar ao objetivo final de ter todas as funcionalidades ativas e executáveis.

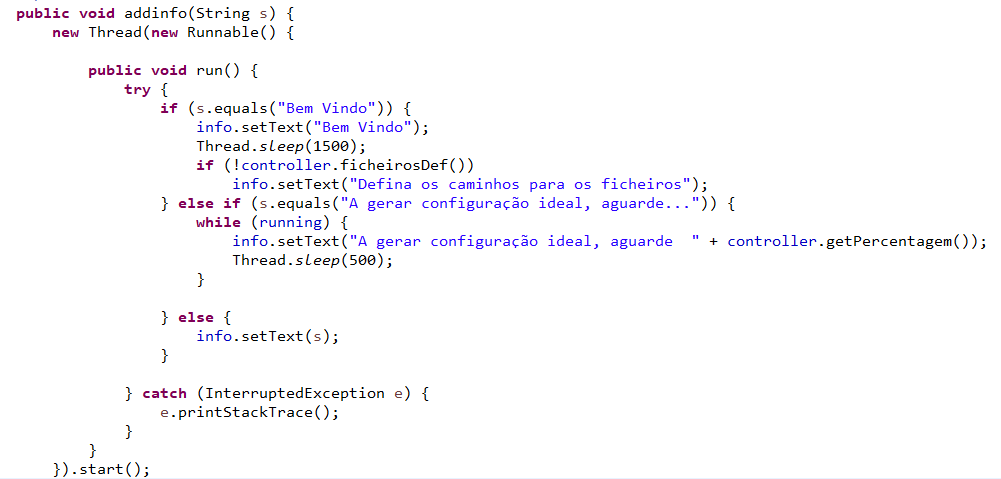
**Sprint 3**

* **Otimização da procura dos pesos de cada regra**

A função que procurava os pesos de cada regra consistia em percorrer cada regra de cada email e percorrer a lista de regras ate encontrar uma igual juntamente com o seu peso correspondente.

Para o calculo do algoritmo esta função (getPeso) era executada 500 milhões de vezes...

Assim, cada email contem um vetor com os índices das suas regras de forma a que seja possível aceder diretamente à posição no vetor rules que corresponde à regra que pretendemos saber o peso.

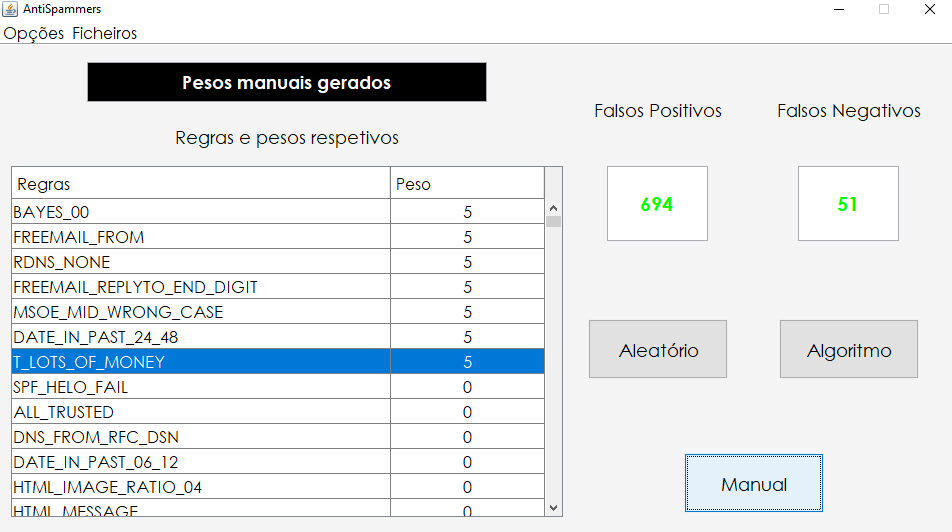
* **Thread autónoma para info**
* **Escolha de caminhos dos ficheiros por defeito no directório do projeto**
* **Permitir ao utilizador gerar uma configuração ideal utilizando o algoritmo NSGA-II**

**Sprint Retrospective 3:**

Neste Sprint tudo correu bem mais uma vez, tirando algumas duvidas iniciais em como implementar a função addinfo(), mas que depois de ultrapassada essa barreira, conseguimos chegar ao final com a função a correr corretamente.

No próximo Sprint vamos continuar a programar a GUI analisar que detalhes faltam para termos tudo a funcionar como é pedido para este projeto ficar finalizado.

**Sprint 4**

* **Comentar código**
* **Permitir a edição da configuração aleatória**
* **Apresentar os resultados da configuração ideal e atualizar o ficheiro rules.cf**



* **Programação da GUI**

**Sprint Retrospective 4:**

Este trata-se do Sprint final, tudo correu bem neste sprint trata-se de finalizar o trabalho, acabar alguns detalhes da parte da programação da GUI e acrescentar alguns detalhes na janela da mesma.

No final chegámos ao objetivo de ter o projeto finalizado com mérito, ao longo dos Sprints não tivemos muitas duvidas alem de alguns detalhes nas funções das threads, mas conseguimos finalizar antes de cada Sprint Retrospective os objetivos.