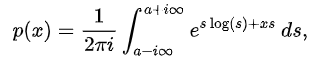
Neste trabalho foi aplicada a distribuição de Landau, pelo que foi feito o fit de Landau nos histogramas para que estes apresentassem uma função obtida refletindo a distribuição de Landau sobre o eixo.

A teoria de Landau consiste numa teoria geral de transições de fase contínuas (ou seja, de segunda ordem), mas também pode ser utilizada como modelo quantitativo para transições descontínuas (ou seja, de primeira ordem). A distribuição de Landau é um caso particular de distribuição estável.

Landau definiu a função de densidade de probabilidade pelo integral complexo:



onde *a* é um número real positivo arbitrário, o que significa que o caminho de integração pode ser qualquer paralelo ao eixo imaginário, intersetando o semi-eixo positivo real, e o log refere-se ao logaritmo natural.

No caso da sua aplicação em ROOT, é definido da seguinte forma:

***double*** *ROOT::Math::landau\_pdf(double x, double xi = 1, double x0 = 0)*

Com os seguintes parâmetros: x como argumento, xi como parâmetro largura e x0 como parâmetro de localização.

Foi também aplicada a linguagem de programação UML (Unified Modeling Language), uma norma emergente para a modelação de software orientado para objectos, que consiste numa linguagem de modelagem que possibilita a representação de sistemas de forma padronizada, com objetivo de proporcionar uma melhor compreensão de pré-implementações. Esta resulta da convergência das notações de três dos principais métodos orientados a objetos: OMT (James Rumbaugh), OOSE (Ivar Jacobson) e Booch (Grady Booch). A diagramação em UML permite estabelecer uma linguagem visual comum no complexo mundo do desenvolvimento de software. Neste trabalho, consideramos que os diagramas UML que melhor representam as implementações requeridas são os seguintes:

Uma imagem com texto, eletrónica, captura de ecrã

Descrição gerada automaticamente

Figura 1 – Activity Diagram (UML)

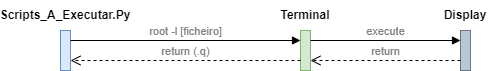


Figura 2 - Sequence Diagram (UML)

[ficheiro] – ficheiros a executar por *root - l* :

* *Histogramas\_Dep\_Event.C*
* Pioes\_Muoes\_Outras.C
* Histogramas\_ETotal\_Particula.C
* Histogramas\_DistXY\_Detetor.C
* Histogramas\_DistXY\_Detetor\_Particula.C
* Histogramas\_DT\_H\_Detetor.C
* Histogramas\_MZ\_MP.C
* Histogramas\_MZ\_P\_PS.C