Manual do Desenvolvedor

BrainLight

**Equipa LGP 5A**

**BrainLight**

**Developers**

André Pinheiro

David Azevedo

João Monteiro

José Lima

Luís Natividade

Luís Pinto

**MM Delegates**

Nerea Castro

Simão Pereira

**Designers**

Diana Magalhães

Mariana Almeida

**Cliente**

INOVA+

**ÍNDICE**

[1. Introdução 3](#_Toc452757193)

[BrainLight 3](#_Toc452757194)

[2. Pré-requisitos 3](#_Toc452757195)

# 1. Introdução

A **BrainLight** é uma *framework* desenvolvida para computadores Windows. Este manual destina-se a programadores com conhecimentos de Java e, idealmente, familiaridade com os SDK dos dispositivos implementados ou de outros que pretendam implementar.

## BrainLight

A **BrainLight** é uma *framework* que lê os dados fornecidos pelos dispositivos EEG compatíveis (numa fase inicial, apenas o são o NeuroSky Mindset e Emotiv Epoc) e os disponibiliza num formato unificado através de uma API que pode ser utilizada por outras aplicações. Para além disso, também inclui uma interface gráfica para a visualização dessas informações, bem como funcionalidades adicionais como análises às ondas e gravação e leitura de um histórico.

# 2. Pré-requisitos

A **BrainLight** foi desenvolvida com a versão 8 para 64 bits do Java SE Development Kit. No desenvolvimento do projeto é necessário incluir todos os módulos que a constituem: BrainLightFW (módulo de gestão e lógica), FW (interface gráfica), Analysis (cálculos de análises às ondas), XLS writer (funcionalidades de histórico). Os módulos NeuroSky\_FW e NeuroSky\_Lib são necessários para a leitura do dispositivo NeuroSky e o módulo Emotiv é o módulo equivalente para esse dispositivo.

O desenvolvimento foi feito recorrendo aos IDEs IntelliJ IDEA e Eclipse em Windows e em Linux.

# 3. Software relevante

Para o desenvolvimento da **BrainLight** foram necessários os SDK dos dispositivos NeuroSky Mindset e Emotiv Epoc. Estes encontram-se no módulo “Devices”, e são necessários para comunicar e ler a informação dos dispositivos.

# 4. API

//TODO

# 5. Processo de criação de módulos

## 5.1 Adicionar um dispositivo

Comece por colocar na pasta “Devices” o conjunto dos ficheiros fornecidos pelo SDK do dispositivo. Depois, na pasta “Logic”, crie um novo módulo com o nome do seu dispositivo.

Nesse módulo que criou terá de criar funções que vão depois comunicar com o “MainModule” por nós criado; se quiser pode consultar como exemplo os ficheiros criados para o Emotiv Epoc e Neurosky Mindset.

O MainModule requere que lhe seja enviada informação no seguinte formato: é um array de arrays......

//TODO: formato + interface

O seu novo dispositivo terá de ser incluído também na interface! Para isso, terá de ir ao módulo “Interface”. Na classe Main, deverá incluir uma nova função de “launch”, igual às já construídas, e incluir também o nome do dispositivo na ComboBox. //TODO

## 5.2 Criar novas análises

As análises encontram-se no ficheiro “Calculations.java”, no módulo “Analysis” (BrainLight/Logic/src/Analysis/). Nesse módulo encontram-se dois outros ficheiros, “Tests.java” e “TypesOfCalculations.java”.

Para acrescentar uma análise é necessário começar por acrescentar um novo elemento ao enum “TypesOfCalculations”, no ficheiro homónimo, com um nome diferente dos usados até esse momento e que descreva a análise que está a ser implementada.

Depois disso, aconselha-se vivamente a criação de testes unitários no ficheiro “Tests.java”. Para isso podem-se seguir os exemplos dados: nos arrays “values” (para valores únicos) e “valuesXY” (para tuplos de valores) encontram-se já alguns exemplos de dados que poderão ser usados. Se for necessário poder-se-ão acrescentar novos elementos a esses arrays ou criar novos. No array TypesOfCalculations será preciso adicionar pelo menos um array no final com o novo elemento criado no enum anteriormente (mas poderão ser criados arrays com mais do que um tipo de cálculo).

Para fazer os testes, é preciso chamar o construtor da classe Calculations. Esse construtor encontra-se “overloaded”: há duas opções de o invocar. O primeiro argumento pode ser um array de floats (para a situação de haver apenas uma dimensão de variáveis) ou um array de arrays de floats (no caso de as variáveis serem bidimensionais). O segundo argumento é sempre um array de enums “TypesOfCalculations”. O programa vai calcular apenas as análises especificadas nesse(s) enum(s). Após invocar o construtor, basta chamar a função “getResult()” para obter uma HashMap de enums (iguais aos enviados) e o valor correspondente a cada um (em float).

Para adicionar a nova análise no ficheiro “Calculations”, basta acrescentar o novo enum no switch presente na função “calc()” dessa classe, bem como uma função nova que calcule essa nova análise e que seja incluída nesse switch. A média é calculada sempre automaticamente devido a ser usada extensivamente em grande parte das análises e o seu valor encontra-se na variável global “mean”.

## 5.3 Implementar novas funcionalidades

Sugerimos que novas funcionalidades que queira implementar na aplicação sejam desenvolvidas dentro do módulo “Logic”, paralelamente ao histórico e análises já desenvolvidos. Depois de implementar uma nova funcionalidade deverá criar funções no MainModule que a utilizem, de modo a poder depois chamar essas funções na Interface.