

MINDSEYE: PROTÓTIPO DE SISTEMA PARA MONITORAMENTO DE ESTÍMULOS CEREBRAIS

Aluno: Luccas de Souza Silva
Orientador: Dalton Solano dos Reis

Roteiro

- Motivação
- Objetivos
- Fundamentação Teórica
- Trabalhos Correlatos
- Especificação
- Ferramentas utilizadas
- Implementação
- Operacionalidade
- Resultados
- Conclusões
- Extensões
- Demonstração

Motivação

- Disponibilidade do equipamento de eletroencefalografia (EEG)
- Potencial de uso de equipamento de EEG na psicopedagogia clínica
- Potencial de melhoria nas avaliações de pacientes
- Necessidade de sistemas para a área

Objetivo Geral

- Protótipo de sistema para monitoramento de estímulos cerebrais.

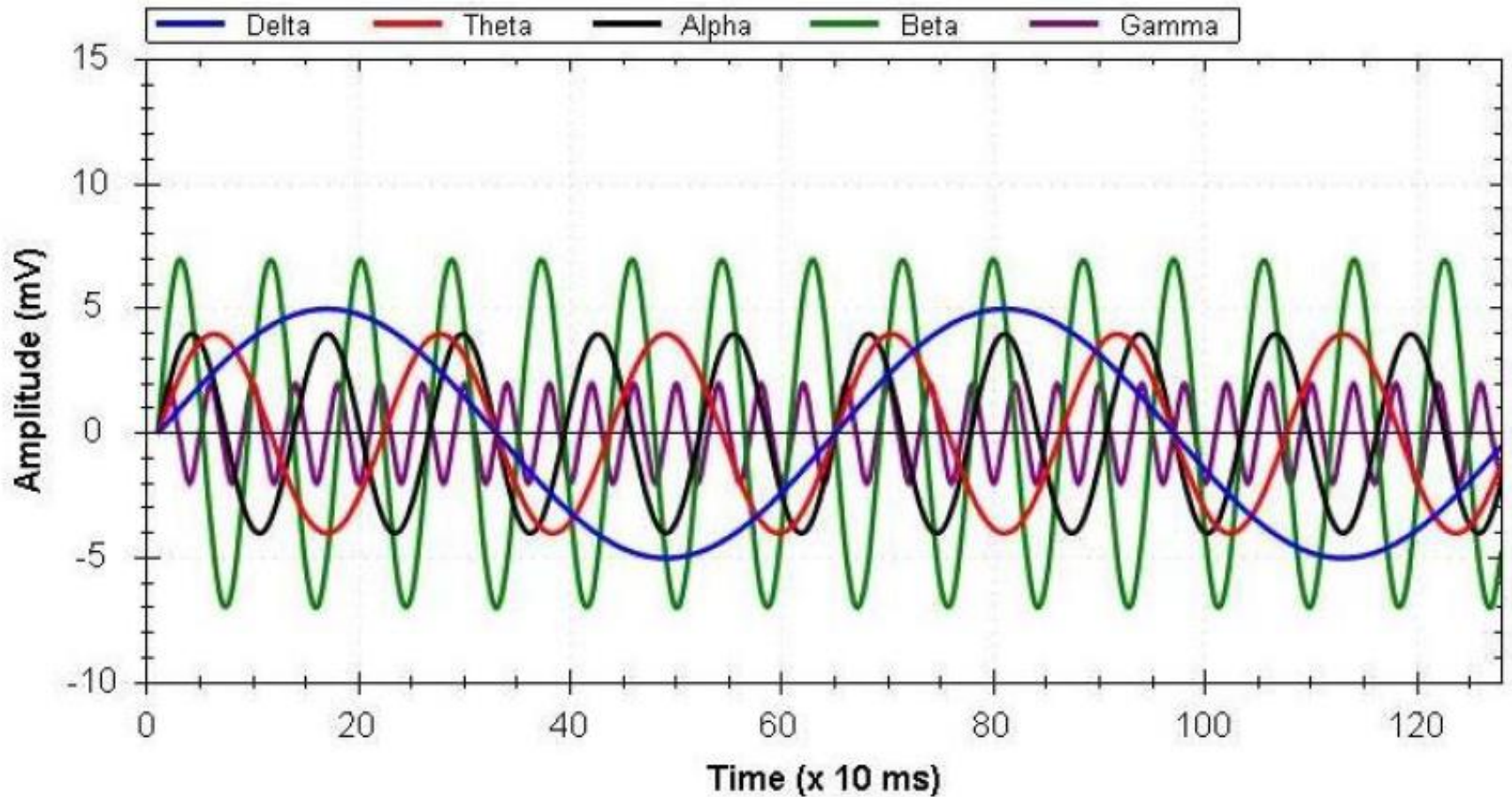
Objetivos Específicos

- Permitir a coleta de dados de pacientes, especialistas e de atendimentos realizados no processo de atendimento psicopedagógico;
- Permitir gravar, durante uma sessão de atendimento, dados de uma câmera de vídeo e dados do equipamento de EEG MindWave Mobile;
- Disponibilizar os dados referentes à gravação de uma sessão, incluindo os dados captados pelo equipamento de EEG quando utilizado

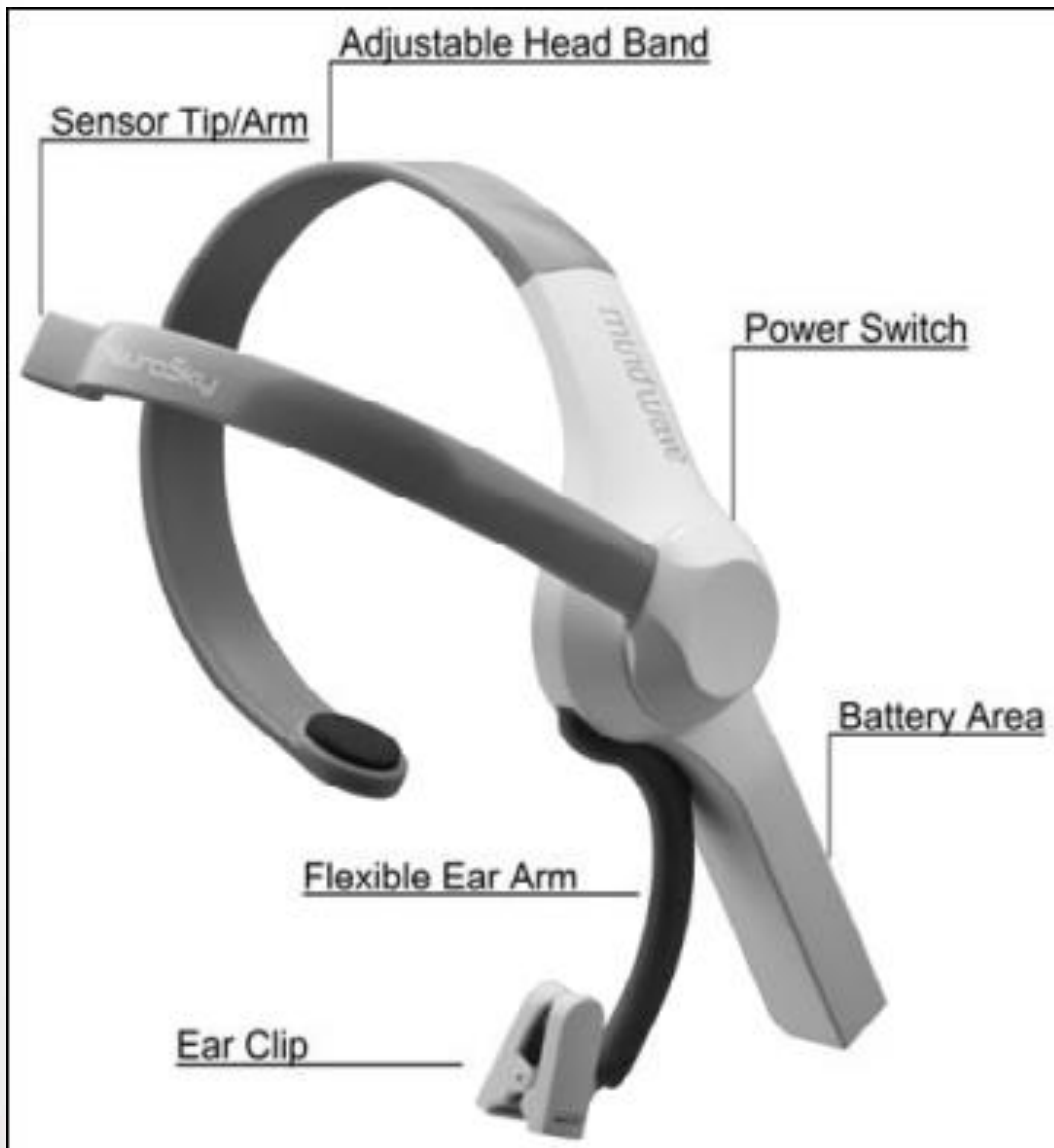
Psicopedagogia e atendimento clínico

- Código de Ética do Psicopedagogo (ABPp, 2011)
- Processo de aprendizagem
- Multidisciplinar
- Avaliações e atendimentos

Estímulos cerebrais e eletroencefalografia

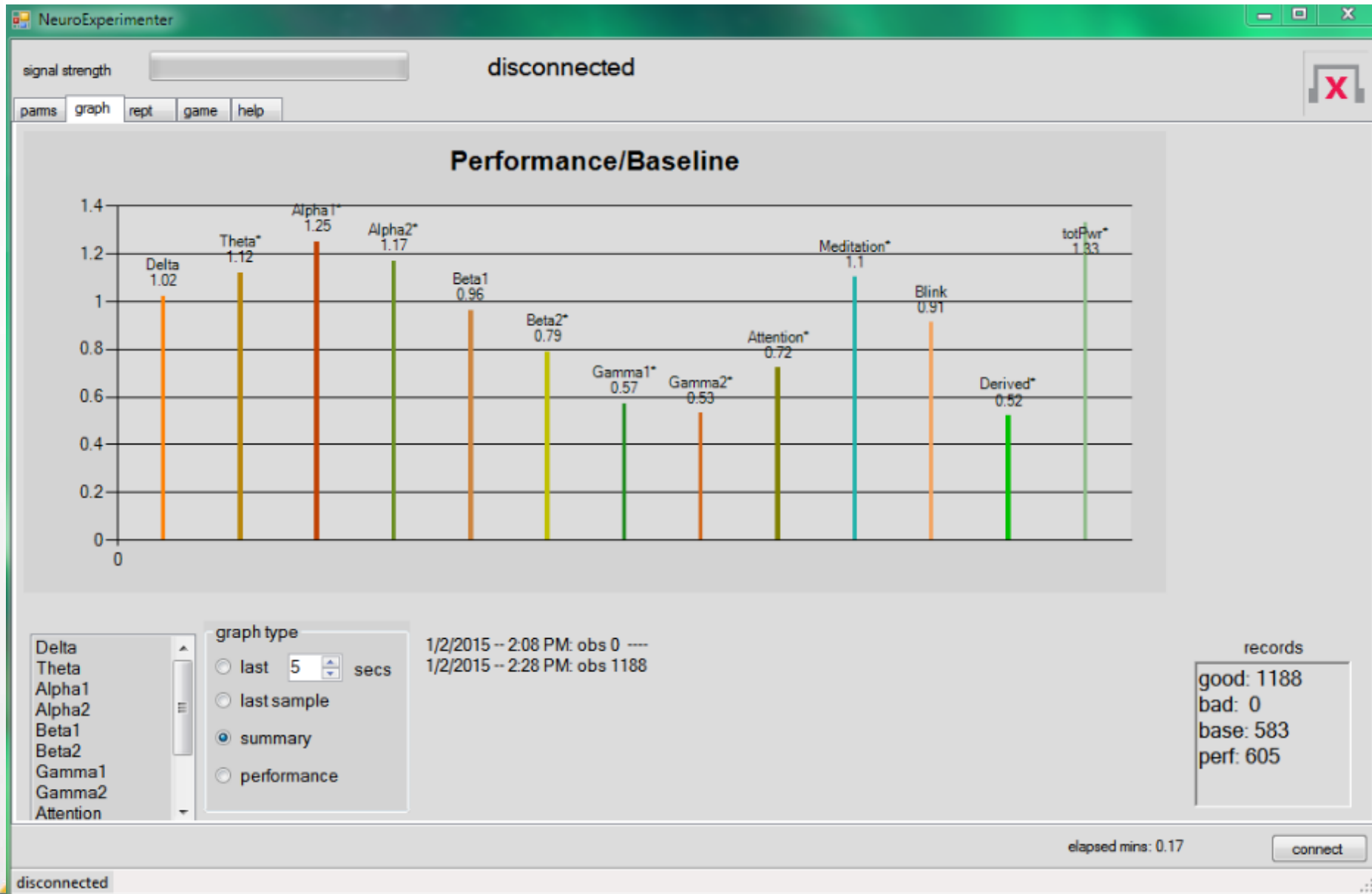


MindWave



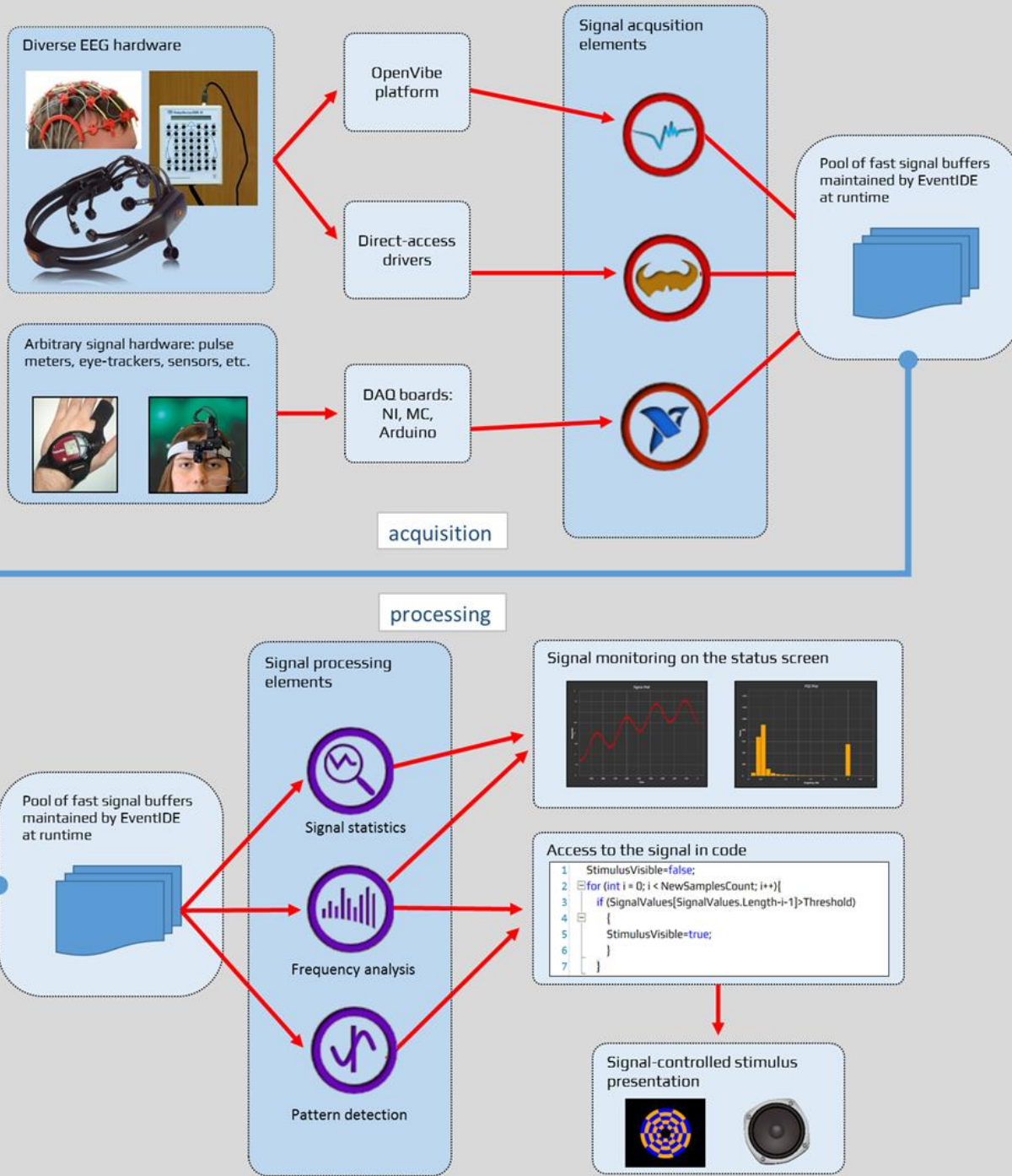
- Estímulos cerebrais
- Nível de atenção
- Nível de relaxamento
- Sensor de piscadas

NeuroExperimenter (Mellender, 2014)



EventIDE

(OkazoLab, 2012)



Paradigm (Perception Research Systems, 2016)

Design View Trial Table View

Events

- Pointer
- Text
- Feedback
- Rich Text
- Audio
- Jump
- Script
- Multimedia
- Image
- Movie
- Pause
- Self-Paced
- Rating Scale
- Block

Unused Events

Response Devices

Ports




Image Element

Name	ImageElement2
Image	
ImageLocation	\\Stimuli\kitten.bmp
Image Element Properties	
Alignment	Center Vertical
BackColor	LightSlateGray
Clickable	True
Location	121, 234
Size	300, 300
Visible	True

Responses:

- Keyboard
- Microphone

Commands:

- Send Marker1

Requisitos funcionais

- Permitir que o usuário cadastre pacientes
- Permitir que o usuário cadastre especialistas
- Permitir que o usuário cadastre sessões para um paciente, vinculando a sessão a um especialista cadastrado
- Permitir a gravação de vídeo para uma sessão
- Permitir captar ondas cerebrais de um usuário através do equipamento de eletroencefalografia MindWave Mobile
- Retornar ao usuário as informações da sessão gravada, incluindo gravação de vídeo e ondas cerebrais captadas

Requisitos não funcionais

- Ser desenvolvido na linguagem de programação C-Sharp
- Utilizar o ambiente de desenvolvimento Visual Studio

Casos de Uso

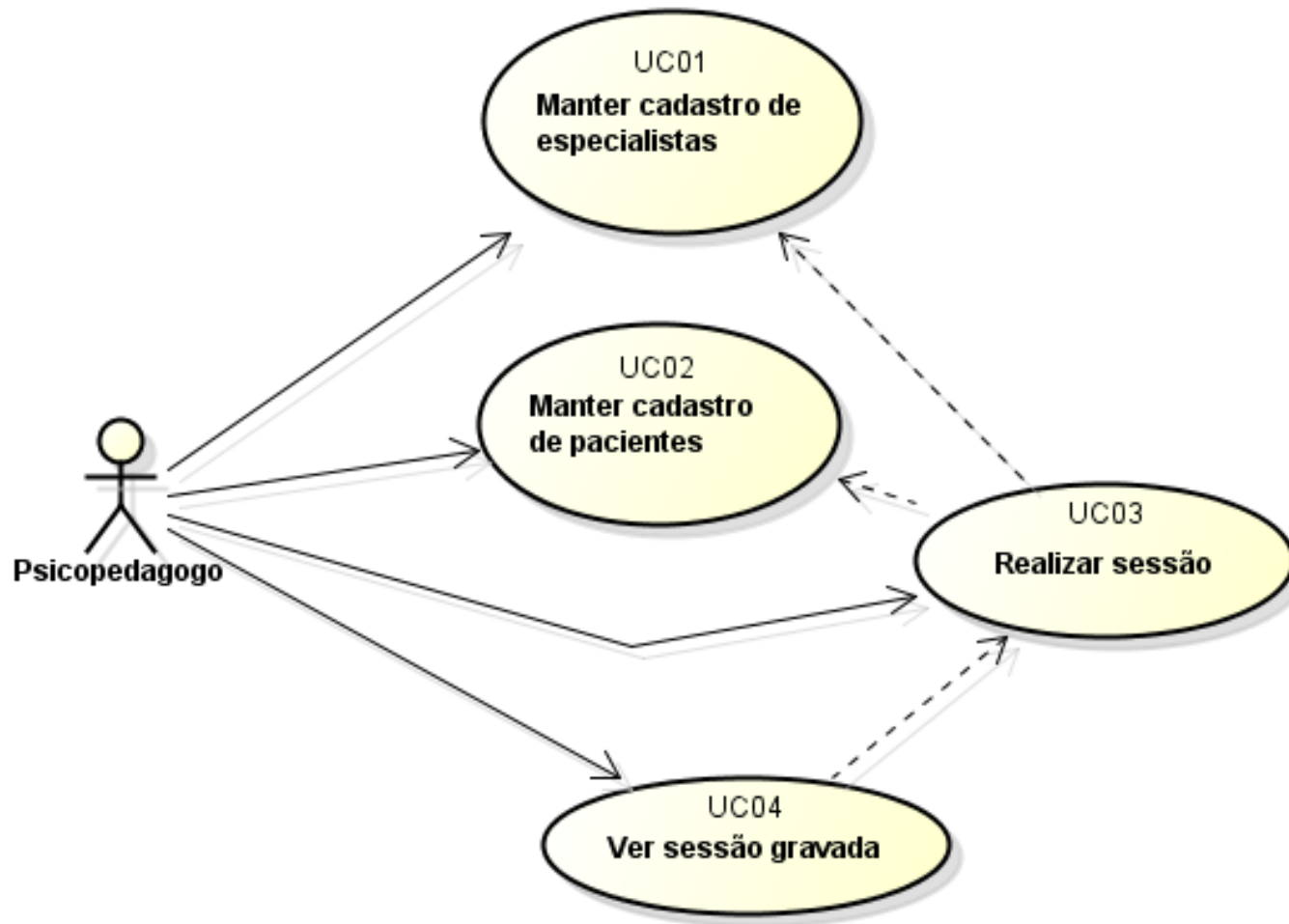
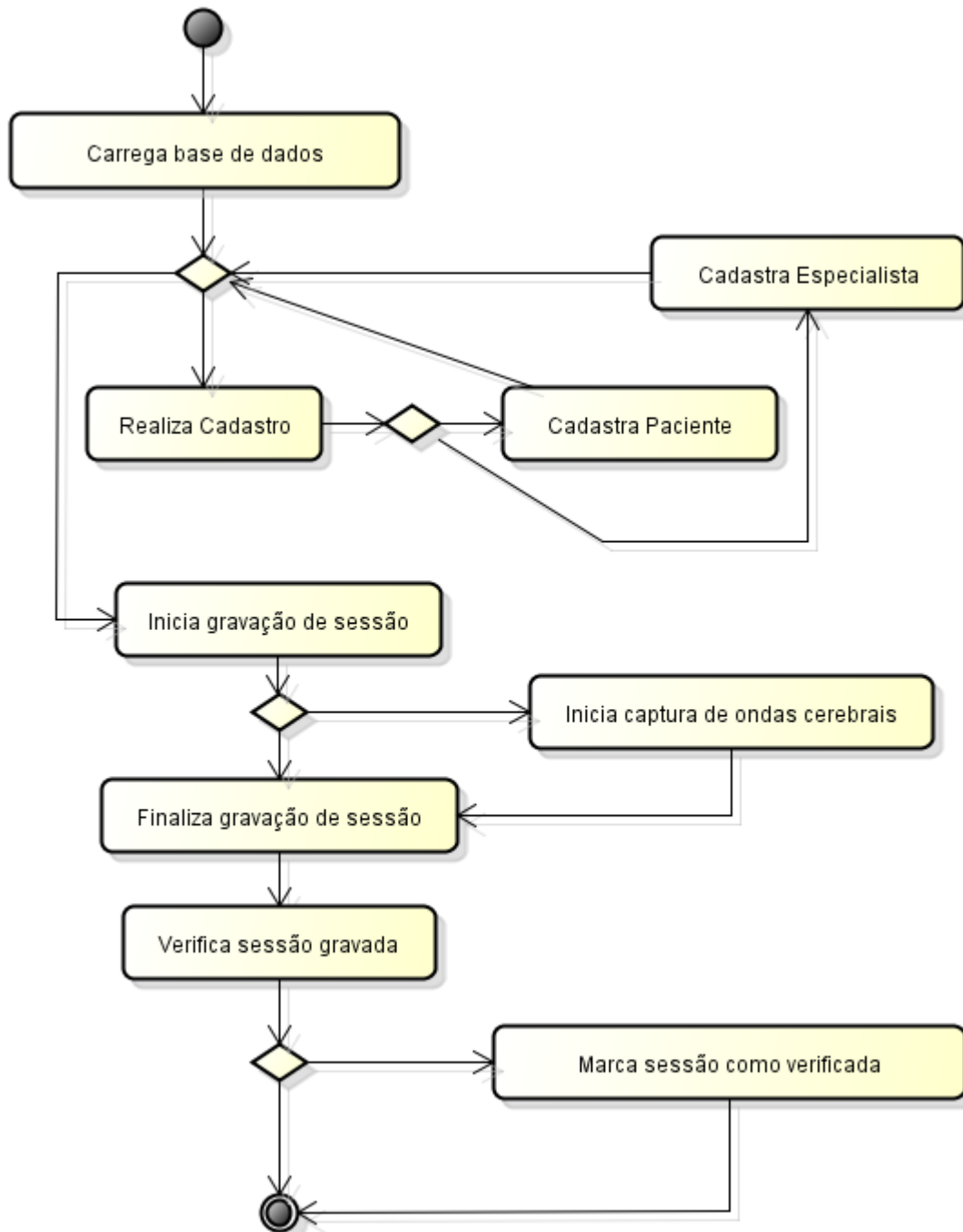
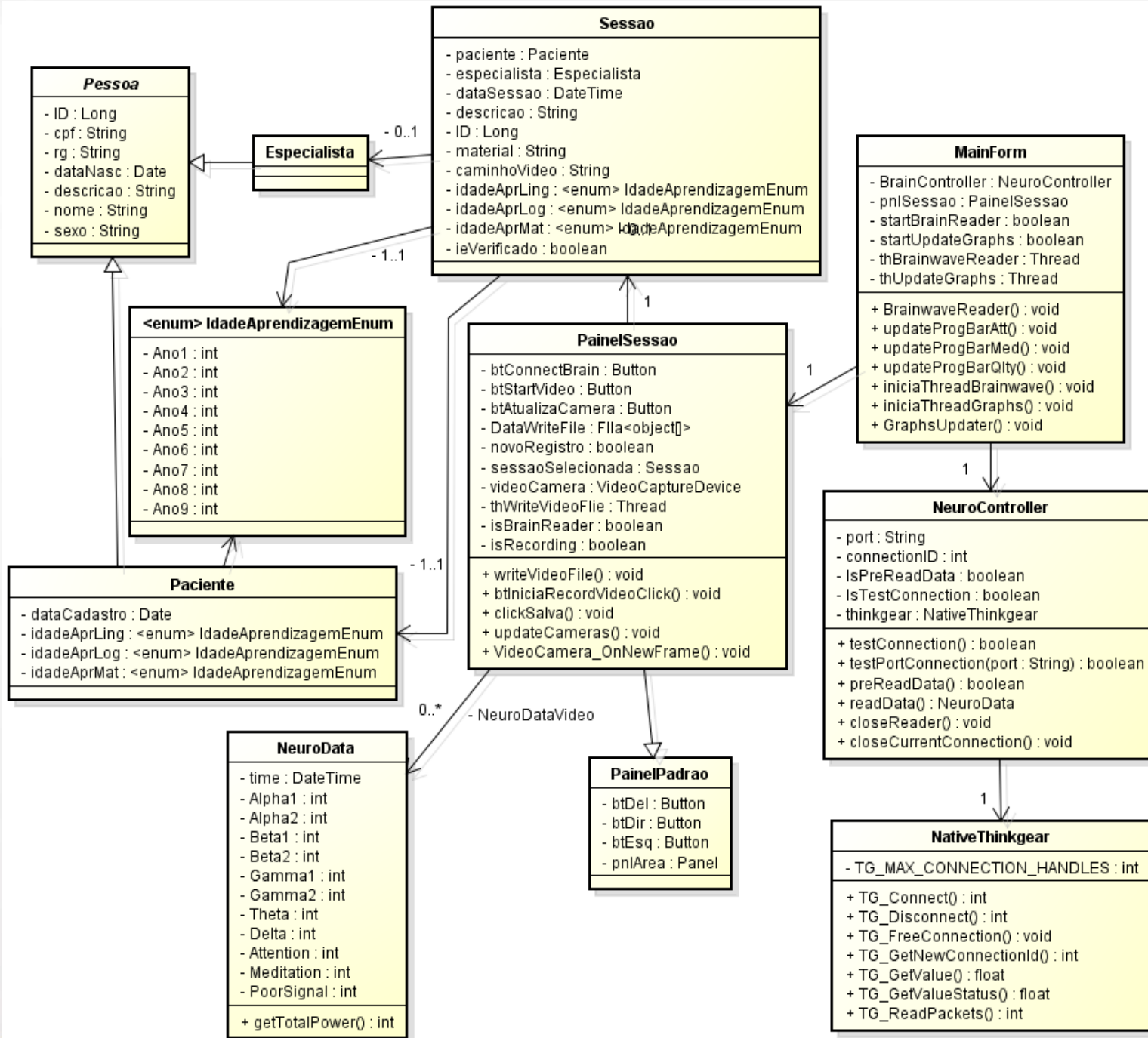


Diagrama de Atividade



Gravação cena



Ferramentas utilizadas



NeuroController

```
public class NeuroController
{
    private string Port;
    private NativeThinkgear thinkgear = new NativeThinkgear();
    private int ConnectionID;

    private bool IsPreReadData;
    private bool IsTestConnection;

    Stopwatch sw = new Stopwatch();
    sw.Start();
    while (sw.ElapsedMilliseconds < 10000)
    {
        errCode = NativeThinkgear.TG_ReadPackets(connectionID, 1);
        if (errCode == 1)
        {
            int[] values = new int[11] {0,0,0,0,0,0,0,0,0,0,0};

            if (NativeThinkgear.TG_GetValueStatus(connectionID, NativeThinkgear.DataType.TG_DATA_POOR_SIGNAL) != 0)
            {
                values[0] = (int)NativeThinkgear.TG_GetValue(connectionID, NativeThinkgear.DataType.TG_DATA_POOR_SIGNAL);
            }

            if (NativeThinkgear.TG_GetValueStatus(connectionID, NativeThinkgear.DataType.TG_DATA_ATTENTION) != 0)
            {
                values[1] = (int)NativeThinkgear.TG_GetValue(connectionID, NativeThinkgear.DataType.TG_DATA_ATTENTION);
            }

            if (NativeThinkgear.TG_GetValueStatus(connectionID, NativeThinkgear.DataType.TG_DATA_MEDITATION) != 0)
            {
                values[2] = (int)NativeThinkgear.TG_GetValue(connectionID, NativeThinkgear.DataType.TG_DATA_MEDITATION);
            }
        }
    }
}
```

```

public void BrainwaveReader()
{
    startBrainReader = true;
    NeuroDataQueue = new Queue<NeuroData>();
    GraphDataQueue = new Queue<NeuroData>();
    bool[] toReadData = new bool[11] { true, true, true,
        true, true, true, true, true, true, true };
    bool sleep = false;

    while (startBrainReader) {
        NeuroData nr = BrainController.readData(toReadData);
        if (nr != null && nr.getTotalPower() > 0) {
            sleep = false;
            lock (NeuroDataQueue) {
                NeuroDataQueue.Enqueue(nr);
            }
            lock (GraphDataQueue) {
                GraphDataQueue.Enqueue(nr);
            }
            if (pnlSessao.isRecording) {
                pnlSessao.addNeuroDataVideo(nr);
            }
        } else {
            sleep = true;
        }
        if (sleep) {
            Thread.Sleep(1);
        }
    }
    BrainController.closeReader();
    NeuroDataQueue = null;
    GraphDataQueue = null;
}

```

Verifica estímulos

- Verifica estímulos em segundo plano
- Enfileira dados lidos
- Fecha conexão ao finalizar

VideoPlayer

- Controla frames
- tempo de vídeo

```
private void VideoProcess() {
    Accord.Video.FFMPEG.VideoFileReader reader = new Accord.Video.FFMPEG.VideoFileReader();
    reader.Open(this.videoURL);
    int fps = (int) reader.FrameRate.Value;
    double videoTimeSeconds = reader.FrameCount / reader.FrameRate.Value;
    startVideoTimeSeconds((int) Math.Floor(videoTimeSeconds));
    setTrackBarMax((int) reader.FrameCount+1);
    setTrackBarPosition(0);
    int i = 0;
    Bitmap frame = reader.ReadVideoFrame(i);
    while ((this.StartVideo && frame != null) || (TrackBarScrollValue >= 0)) {
        if (!this.PauseVideo) {
            if(TrackBarScrollValue >= 0) {
                i = TrackBarScrollValue;
                TrackBarScrollValue = -1;
            } else {
                i = getTrackBarPosition();
            }
            frame = reader.ReadVideoFrame(i);
            if(frame != null) {
                setDisplayImage(frame);
                Thread.Sleep(0020);
                setTrackBarPosition(i + 1);
            }
            double segundos = i / fps;
            setVideoTime((int) Math.Floor(segundos));
        }
    }
    this.isRunning = false;
    this.StartVideo = false;
    this.PauseVideo = false;
    TrackBarScrollValue = 0;
    setTrackBarPosition(0);
    setVideoTime(0);
    setDisplayImage(reader.ReadVideoFrame(0));
    setBtStartPauseImage(PlayImage);
    reader.Close();
}
```

Vídeo para Gráficos

```
public void GraphsUpdater() {
    while (startUpdateGraphs) {
        NeuroData nr = null;
        lock (this.GraphDataQueue) {
            if (this.GraphDataQueue != null
                && this.GraphDataQueue.Count > 0) {
                try {
                    nr = this.GraphDataQueue.Dequeue();
                } catch (Exception ex) {
                    nr = null;
                }
            }
        }
        if (nr != null) {
            if(nr.Meditation <= 100) {
                this.updateProgBarMed(nr.Meditation);
            }
            if (nr.Attention <= 100) {
                this.updateProgBarAtt(nr.Attention);
            }

            int p = (int) (((nr.PoorSignal*100)/255)*2);

            if (p > 100 && (nr.Meditation > 15 || nr.Attention > 15)) {
                p -= 100;
            }

            if (p > 200) {
                p = 200;
            } else if(p < 0) {
                p = 0;
            }

            this.updateProgBarQty(200 - p);
            if (this.formGraph != null && this.formGraph.isGraphReady) {
                this.formGraph.updateGraphs(nr);
            }
        }
    }
    if (!isClosing) {
        this.enableBtDetalheGraph(false);
    }
}
```


Operacionalidade da Implementação

MindsEye

Pacientes

ID	Nome	Data Nasc	Idade Original	Idade Atual	Lingu	
+	3	Paciente A	19/01/2008	10 a. 5 m.	10 a. 5 m.	6o An
+	2	Paciente B	03/07/2007	11 a. 0 m.	11 a. 0 m.	7o An
+	1	Paciente C	04/03/2000	18 a. 4 m.	18 a. 4 m.	4o An

Sessões




ID: 1
Material:
Data Sessao: 05/07/2018 07:41
Especialista: Especialista A
Atualizar Camera: Integrated Webcam

Finalizar
Conectar

Qualidade do Sinal

Forte



Fraco

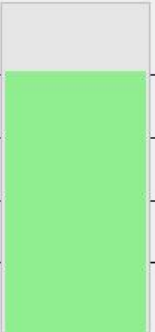
Atenção

Alto


Elevado

Normal

Reduzido



Calma/Relaxamento



Detalhes

Especialistas Config

Operacionalidade da Implementação

Configurações

Informações técnicas

Local dos arquivos: C:\Users\Luccas\Documents

Atualizar Camera padrão: Integrated Webcam

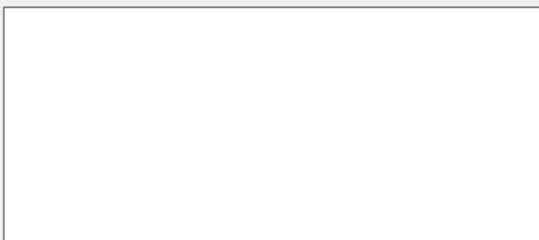
Informações do estabelecimento

Nome:

Endereço:

Telefone:

Imagem para Logo:



Salvar

Especialistas

Voltar

Especialistas

Salvar

ID: 1

Nome:

Data Nasc: 11/07/2018

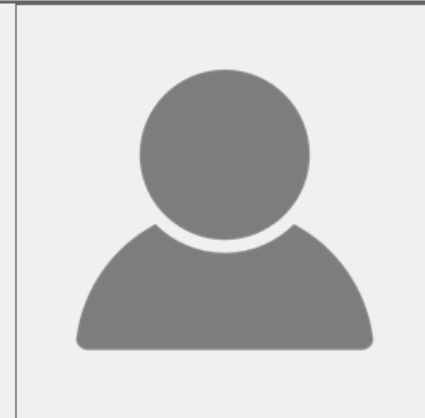
CPF:

RG:

Sexo: Feminino Masculino Outro

Observações:

Observações text area



Alterar Foto

Resultados e Discussões

- Teste com uma psicopedagoga
- Comparação com correlatos:

	Protótipo	Mellender (2014)	OkazoLab (2012)	Perception Research Systems (2016)
Permite scripts customizados	Não	Não	Sim	Sim
Exibe estímulos captados	Sim	Sim	Sim	Sim
Controle de Áudio	Não	Sim	Sim	Sim
Gravação de Vídeo	Sim	Não	Sim	Sim
Ambiente de utilização	Clínicas	Clínicas e Pesquisas	Pesquisas	Pesquisas

Conclusões

- Objetivo geral e objetivos específicos foram atingidos;
- É um protótipo, não um produto;
- Motivou discussões, críticas e ideias;
- Projeto de base:
 - Interesse da profissional entrevistada;
 - Novos trabalhos

Extensões

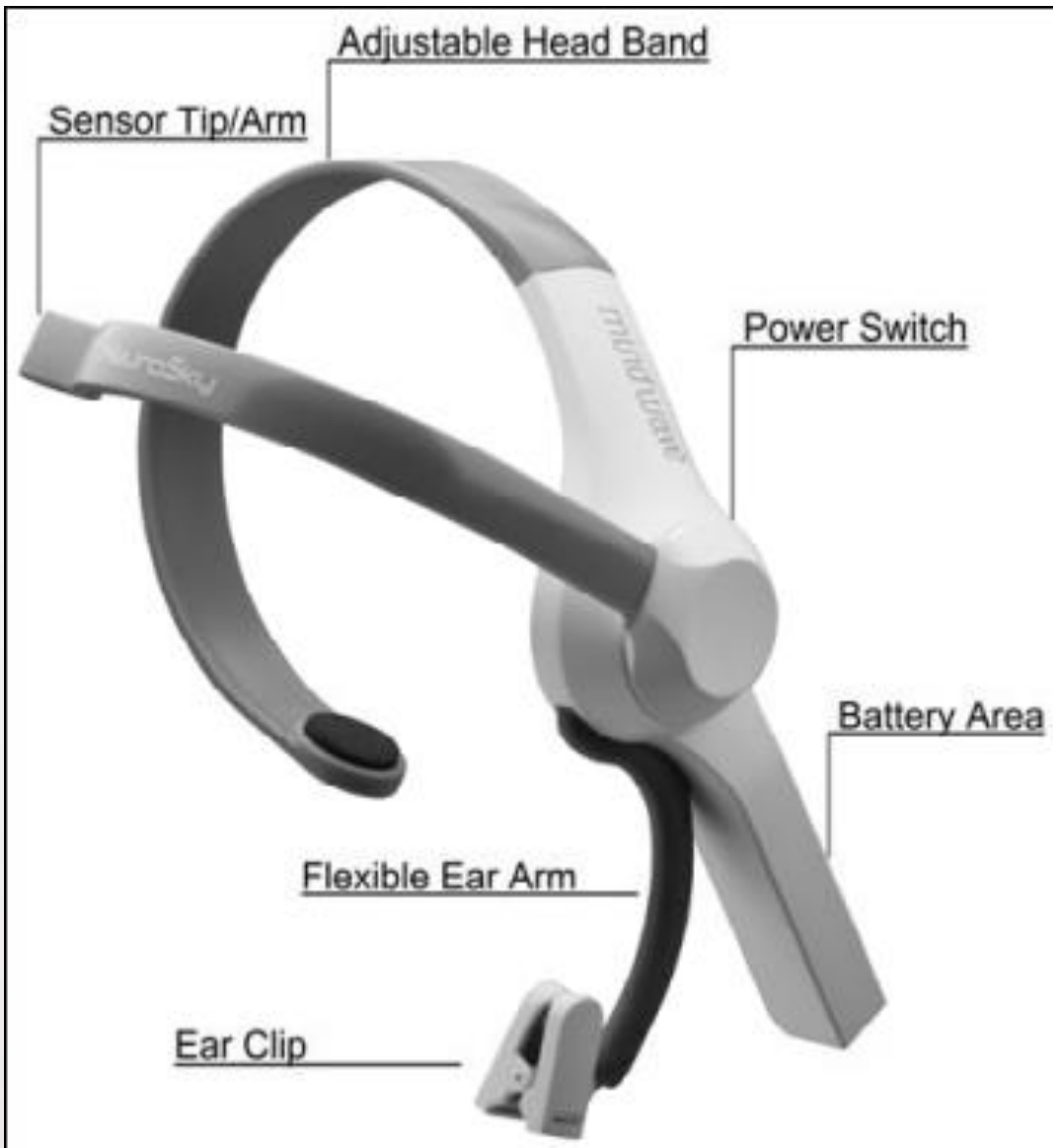
- Melhoria de performance e usabilidade;
- Mais informações no cadastro de paciente;
- Mais informações na gravação: sessão;
- Implementar gravação de áudio com vídeo.

Extensões

- Opção de marcar momentos em uma gravação;
- Integração com outros equipamentos de EEG;
- Relatório geral de paciente e desempenho;
- Modificar processo de codificação e decodificação da gravação.

Demonstração

MindWave Mobile



- Empresa NeuroSky
- \$100 (sem frete)
- Bluetooth
- 1 Bateria AAA (8hrs)
- Win (XP/7/8/10)
- Mac (OSx > 10.8)
- iOS (> 8)
- Android (> 2.3)