N2 - FLUTTER

Equipe: Guilherme Halter Nunes; João Vitor Bagatoli.

Análise de Requisitos

Visão e Escopo

O **Sistema Automatizado de Chamada** é um aplicativo móvel (Android) desenvolvido em Flutter, cuja principal missão é modernizar a verificação de presença em sala de aula, eliminando a necessidade de intervenção manual do professor. A visão do projeto é entregar uma solução que garanta que a presença registrada se baseie na **proximidade física** do aluno com o dispositivo do professor, mitigando fraudes.

Os objetivos centrais incluem a **automatização total** da chamada, programando o disparo de 4 rodadas de verificação por aula através de um temporizador interno, e a **garantia de presença** por meio de um mecanismo de proximidade, como o **Bluetooth Low Energy (BLE)**. Crucialmente, o aplicativo deve operar de forma autônoma, sem depender de infraestrutura externa, servidores institucionais, IP fixo, ou hardware adicional, usando apenas o que alunos e professores já possuem (seus próprios dispositivos móveis).

Para esta fase inicial (N2), o escopo está estritamente focado na entrega de um **protótipo funcional** no emulador Android. Isso envolve o desenvolvimento das telas navegáveis e a lógica básica para a temporização das rodadas e o *feedback* de status de proximidade, com os dados de presença sendo mantidos apenas na memória durante a sessão da aula. Funcionalidades como autenticação de usuário serão simuladas com simples botões de seleção de perfil. A persistência permanente de dados (banco de dados) está fora do escopo atual. Contudo, o projeto deve prever a estrutura de dados de presença ($\text{Aluno} \times \text{Rodada}$) para a futura exportação do relatório consolidado em formato **CSV** (requisito para a N3), garantindo a escalabilidade da lógica de registro.

StakeHolders e Personas

Os **Stakeholders Primários** são o **Professor da Disciplina** (que define e avalia os requisitos de entrega, como a automatização, a mitigação de burlas e a entrega do protótipo) e a **Equipe de Desenvolvimento**. A **Instituição de Ensino** é um *stakeholder*

secundário que se beneficia indiretamente da eficiência e modernização do registro de frequência.

Para a usabilidade e o design do aplicativo, definimos duas **Personas Chave**:

1. Professor

Objetivo com o App: Iniciar a aula com um único toque e ter a certeza de que a chamada será gerenciada automaticamente a cada rodada, sem interrupções. Ele precisa de um *dashboard* claro que mostre quem está presente ou ausente no momento, garantindo que sua atenção permaneça no conteúdo da aula, e não na administração da frequência.

2. Aluno

Objetivo com o App: Abrir o aplicativo ao entrar na sala, clicar em "Participar da Chamada" e esquecer o assunto. Ela não quer interações manuais repetidas e espera um feedback rápido e claro, informando que sua presença foi detectada pelo dispositivo do professor e registrada com sucesso na rodada atual, sem consumir excessivamente a bateria do seu celular.

Regras de Negócio

R.N001 - Automação da Chamada (4 Rodadas Fixas)

- **DESCRIÇÃO:** O sistema deve executar exatamente quatro (4) janelas de chamada (rodadas) por sessão de aula. Uma vez iniciada a aula, o professor não pode iniciar ou disparar rodadas manualmente.
- **REQUISITO:** O *Timer* interno deve ser o único responsável pelo controle do tempo.

R.N002 - Intervalo Fixo entre Rodadas

- **DESCRIÇÃO:** As rodadas de chamada devem ser espaçadas por um intervalo fixo. A premissa é de aproximadamente 50 minutos entre o início de cada rodada (ex: Rodada 1 em 0 min, Rodada 2 em 50 min, Rodada 3 em 100 min, Rodada 4 em 150 min).
- **REQUISITO:** A variação do intervalo deve ser minimizada para manter a previsibilidade.

R.N003 - Duração da Janela de Registro

- **DESCRIÇÃO:** Cada rodada terá uma janela de tempo curta e definida para o registro de presença, durante a qual o dispositivo do Professor fará o *Advertising* BLE.
- **REQUISITO:** Sugere-se uma janela de **5 minutos** para cada rodada. O registro de presença fora dessa janela é **automaticamente rejeitado**.

R.N004 - Prova de Proximidade (Anti-Burla)

- **DESCRIÇÃO:** Para que a presença de um aluno seja registrada com sucesso em qualquer rodada, o dispositivo do Aluno deve detectar o sinal BLE do Professor com uma **força de sinal mínima aceitável** (RSSI).
- **REQUISITO:** O sinal deve ser forte o suficiente para indicar que o Aluno está fisicamente dentro do raio da sala de aula. Sinais fracos (indicando distância) devem ser rejeitados.

R.N005 - Presença Única por Rodada

- **DESCRIÇÃO:** O aluno só pode ter um único status de presença registrado por rodada de chamada.
- **REQUISITO:** Se um aluno sair e voltar, ele só poderá registrar uma nova presença na próxima rodada automática, impedindo múltiplos registros na mesma janela.

R.N006 - Ausência por Não-Detecção

- **DESCRIÇÃO:** Um aluno será classificado como "Ausente" na rodada se o sistema do Professor não receber a confirmação de presença (devido à falta de proximidade ou não ativação do aplicativo/BLE do Aluno) dentro da janela de registro (R.N003).
- **REQUISITO:** O sistema do Aluno deve estar com o Bluetooth ligado e o aplicativo ativo para realizar o *Scanning*.

R.N007 - Estrutura de Registro de Presença

- **DESCRIÇÃO:** O registro de presença deve ser indexado por uma tripla de chaves para rastreamento: ID do Aluno, Data da Aula e Número da Rodada (1, 2, 3 ou 4).
- **REQUISITO:** O objeto de dados deve incluir o *timestamp* exato da confirmação de presença.

R.N008 - Persistência Temporária de Dados (N2)

- **DESCRIÇÃO:** Nesta fase (N2), os dados de presença da aula em andamento devem ser armazenados exclusivamente na **memória volátil** do dispositivo do Professor.
- **REQUISITO:** Não deve haver dependência de bancos de dados locais (SQLite) ou remotos. Os dados são perdidos ao fechar o aplicativo.
- **STATUS**: Substituído
- **MODIFICAÇÃO**: Dados de alunos, chamadas e registros de presença são persistidos em um banco de dados em nuvem (Supabase).

R.N009 - Previsão do Formato de Exportação CSV

- **DESCRIÇÃO:** Embora a exportação real não seja implementada na N2, o modelo de dados deve ser desenhado para suportar a geração de um relatório CSV consolidado.
- **REQUISITO:** O formato do arquivo deve incluir colunas para RA do Aluno, Nome do Aluno, Data, Rodada e Status (P/A).

Requisitos Funcionais

R.F001 - Seleção de Perfil de Usuário

- O sistema deve permitir que o usuário selecione explicitamente (e de forma simulada) se está usando o aplicativo como **Professor** ou como **Aluno**.
- **STATUS**: Substituído
- **MODIFICAÇÃO**: O sistema agora exige Login e Cadastro de Aluno para acesso, validando as credenciais no Supabase Auth.

R.F002 - Navegação Professor

- O sistema deve exibir uma tela inicial para o Professor com opções claras para **"Iniciar Nova Chamada"** e **"Visualizar Relatório (Simulado)"**.

R.F003 - Navegação Aluno

- O sistema deve exibir uma tela inicial para o Aluno com uma opção clara para **"Participar da Chamada"**.

R.F004 - Exibição do Status da Chamada (Professor)

- Durante uma chamada ativa, o sistema deve exibir claramente a **Rodada Atual** (ex: 3/4) e um **Contador Regressivo** para o início da próxima rodada, indicando a automação (R.N001, R.N002).

R.F005 - Lista de Presença em Tempo Real (Professor)

- O sistema deve exibir, na tela de Chamada Ativa, uma lista atualizada de todos os **alunos que registraram presença** com sucesso na rodada atual.
- **STATUS**: Melhorado
- **MODIFICAÇÃO**: A lista de presença do Professor é atualizada em tempo real utilizando o recurso de Stream do Supabase, reagindo imediatamente aos novos registros.

R.F006 - Feedback de Sucesso (Aluno)

- O sistema deve exibir ao Aluno uma mensagem de **confirmação clara** quando sua presença for detectada via proximidade e registrada com sucesso na rodada em curso.

R.F007 - Feedback de Falha/Erro (Aluno)

- O sistema deve notificar o Aluno quando a presença **não puder ser registrada**, seja por estar **fora da janela de tempo** (R.N003) ou por **não detectar o sinal de proximidade** do professor (R.N004).

R.F008 - Inicialização da Automação

- O Professor deve poder iniciar a sessão de chamada. Ao iniciar, o aplicativo deve **ativar o Timer de Automação** (R.N001) e iniciar a simulação do **Advertising BLE** do Professor.

R.F009 - Simulação de Proximidade (BLE/P2P)

- O sistema deve implementar uma **lógica de simulação** (ou uso real de API BLE, se possível no emulador) para que o dispositivo Aluno detecte a presença do sinal do Professor e o use como critério primário de registro (R.N004).

R.F010 - Validação Temporal do Registro

- O sistema deve garantir que o registro de presença do Aluno seja aceito pelo Professor **apenas** se o evento ocorrer dentro dos limites da janela de tempo da Rodada (R.N003).

R.F011 - Geração de Registro em Memória

- Ao receber uma confirmação válida (proximidade + tempo), o sistema deve **criar e armazenar em memória** um registro de presença contendo o ID do Aluno, a Rodada e o Timestamp (R.N007, R.N008).

R.F012 - Visualização do Relatório da Sessão

- O sistema deve permitir ao Professor visualizar, ao final da chamada, um relatório simples em tela com os resultados consolidados da sessão, mostrando $\text{Aluno} \times \text{Status (P/A)}$ para cada uma das 4 rodadas.
- **STATUS**: Melhorado
- **MODIFICAÇÃO**: A tela de Relatório de Chamada busca dados do banco de dados para gerar um relatório consolidado com a presença por rodada de todos os alunos do dia.

R.F013 - Encerramento da Chamada

- O Professor deve ter a opção de **encerrar a sessão de chamada** antes que o *Timer* automático finalize a 4ª rodada. Ao encerrar, o sistema deve parar o *Advertising* e consolidar os dados registrados.

R.F014 - Previsão do Formato de Dados CSV

- O sistema deve possuir a estrutura de dados necessária para, futuramente, gerar um arquivo CSV com o formato específico: $\text{RA}, \text{Nome}, \text{Data}, \text{Rodada}, \text{Status}$ (R.N009).

Requisitos Não Funcionais

R.NF001 - Latência de Registro de Presença

- **DESCRIÇÃO:** O tempo que o dispositivo do Aluno leva para detectar o sinal do Professor e registrar a presença (dentro da janela de 5 minutos) deve ser o mais rápido possível.
- **META:** O registro de presença deve ser concluído e confirmado na tela do Aluno em no máximo **3 segundos** após a detecção do sinal forte do Professor.

R.NF002 - Estabilidade do Timer de Automação

- **DESCRIÇÃO:** O *Timer* que dispara as 4 rodadas deve ser robusto e preciso, mesmo se o aplicativo estiver em segundo plano por curtos períodos no dispositivo do Professor.
- **META:** A variação no tempo de disparo de cada rodada não deve exceder **± 5 segundos** do horário programado.

R.NF003 - Consumo de Bateria (Aluno)

- **DESCRIÇÃO:** A operação contínua de *Scanning* (busca por BLE) no dispositivo do Aluno não deve drenar excessivamente a bateria.
- **META:** O consumo de bateria pelo aplicativo do Aluno durante uma aula de 4 horas não deve ser superior a **15%** do total da bateria.

R.NF004 - Intuitividade da Interface (Professor)

- **DESCRIÇÃO:** As ações principais do Professor (Iniciar Chamada, Finalizar) devem ser óbvias e exigir o mínimo de cliques.
- **META:** O Professor deve conseguir iniciar a chamada em no máximo **dois cliques** a partir da tela principal.

R.NF005 - Clareza do Status (Aluno)

- **DESCRIÇÃO:** A tela de status do Aluno deve fornecer *feedback* imediato e inequívoco sobre se a presença foi registrada ou se está em busca do sinal.
- **META:** O *feedback* de status (sucesso ou falha) deve ser apresentado em tela com cores e mensagens contrastantes (ex: verde para sucesso, vermelho/amarelo para erro/busca).

R.NF006 - Restrição de Dependência Externa

- **DESCRIÇÃO:** O aplicativo não pode, sob nenhuma circunstância, depender de acesso a servidores da instituição, IP fixo ou hardware especial (ex: *gateways* BLE dedicados).
- **META:** A comunicação deve ser estritamente *peer-to-peer* (ponto-a-ponto) ou de proximidade (BLE) entre os dispositivos móveis.

R.NF007 - Execução em Emulador Android (N2)

- **DESCRIÇÃO:** O protótipo deve ser totalmente funcional e demonstrável no emulador oficial do Android Studio ou VS Code.

- **META:** O *build* deve ser bem-sucedido e o aplicativo deve ser executado sem falhas críticas de *runtime* no emulador (visando API 28+).

R.NF008 - Portabilidade do Código

- **DESCRIÇÃO:** O código Flutter deve ser escrito de forma que, se no futuro as restrições mudarem, ele possa ser compilado para iOS com o mínimo de alterações possível (mantendo-se o mesmo *codebase* para a lógica).
- **META:** A lógica de temporização e a UI devem usar pacotes e APIs Flutter genéricos, isolando qualquer código nativo específico de BLE ou Android (como permissões) em módulos de serviço dedicados

Ameaças e Antifraude (Alto nível)

As ameaças principais são mitigadas da seguinte forma:

1. **Fraude de Proximidade (Spoofing e Revezamento):** A falsificação de sinal à distância (**A.01**) é combatida exigindo uma **força mínima de sinal BLE (RSSI forte - M.01)**. O revezamento de dispositivos (**A.02**) é dificultado pela **janela de registro curta (M.03)** e pela regra de **registro único por rodada (M.02)**. Ataques mais sofisticados, como **Replay Attack (A.03)**, serão futuramente mitigados pelo uso de **tokens criptográficos variáveis (M.04)** no Advertising do professor.
2. **Fraude de Integridade/Automação:** A manipulação do cronograma pelo professor (**A.04**) é evitada pela **Automação Forçada (M.05)** do Timer, que impede disparos manuais. A **Fraude de Horário (A.05)** é mitigada pela **Validação Temporal (M.06)** feita pelo dispositivo do Professor, que usa seu próprio *timestamp* como fonte de verdade para aceitar o registro dentro da janela de 5 minutos. O **registro é indexado por chave única (M.07)** para impedir a duplicação de presenças na mesma rodada (**A.06**).

Critérios de Aceite Objetivos

I. Usabilidade e Configuração Inicial

- **CA.01 - Seleção de Perfil:** A tela inicial permite a seleção simulada de "Professor" ou "Aluno" com **um toque**.
- **CA.02 - Início Rápido (Professor):** O Professor inicia a sessão e ativa o Timer em **no máximo dois cliques**.

II. Funcionalidade e Precisão do Timer

- **CA.03 - Ativação da Automação:** Clicar em "Iniciar Nova Chamada" exibe imediatamente o **contador regressivo** para a Rodada 1.

- **CA.04 - Precisão do Timer:** O Timer dispara o início da Rodada 2 no tempo programado (ex: 50 minutos) com variação máxima de **±5 segundos**, mesmo simulando *background*.

III. Lógica de Registro e Antifraude

- **CA.05 - Janela de Registro:** O registro só é aceito **dentro da janela de 5 minutos** da rodada; fora desse limite, é rejeitado.
- **CA.06 - Prova de Proximidade (Simulada):** A lógica de registro é acionada apenas quando a simulação indicar **sinal forte** do Professor (condição RSSI).
- **CA.09 - Unicidade do Registro:** A tentativa de um Aluno registrar a presença mais de uma vez na mesma Rodada deve resultar em **apenas um único registro** na memória do Professor.

IV. Performance e Feedback

- **CA.07 - Latência de Registro:** O *feedback* de sucesso na tela do Aluno é exibido em **no máximo 3 segundos** após a detecção simulada do sinal.
- **CA.08 - Feedback Claro (Aluno):** O status do Aluno deve ser visualmente claro, usando **cores e mensagens contrastantes** (ex: verde para sucesso).
- **CA.10 - Atualização em Tempo Real:** O nome do Aluno aparece na Lista de Presença do Professor em **no máximo 1 segundo** após o registro de sucesso.

V. Estrutura de Dados e Persistência (Temporária)

- **CA.11 - Consolidação em Memória:** Ao final da sessão, a estrutura de dados em memória deve conter **exatamente 4 status** (P ou A) para cada aluno na turma.
- **CA.12 - Estrutura de Dados (CSV):** O modelo de dados interno possui todos os campos obrigatórios para o futuro CSV: RA,Nome,Data,Rodada,Status e Timestamp.

VI. Restrições Técnicas e Ambientais

- **CA.13 - Execução em Emulador:** O aplicativo deve ser compilado e executado **sem falhas críticas** no emulador oficial do Android Studio (API 28+).
- **CA.14 - Autonomia da Comunicação:** A funcionalidade de registro deve ser demonstrada em um ambiente com **internet desligada** (Wi-Fi/dados móveis), confirmando a comunicação peer-to-peer.

Protótipos de CSV

Estrutura de Colunas

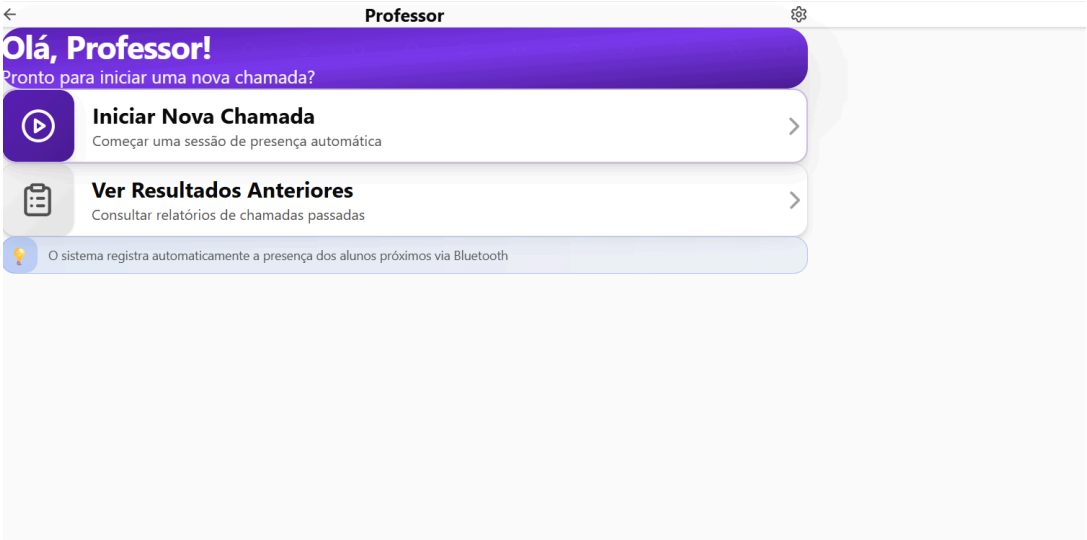
Coluna	Descrição
RA	Registro Acadêmico único do aluno.
Nome	Nome completo do aluno.

Data	Data da aula (AAAA-MM-DD).
Rodada	Número da rodada (1 a 4).
Timestamp	Horário exato do registro da presença (ex: 2025-10-25 08:05:32).

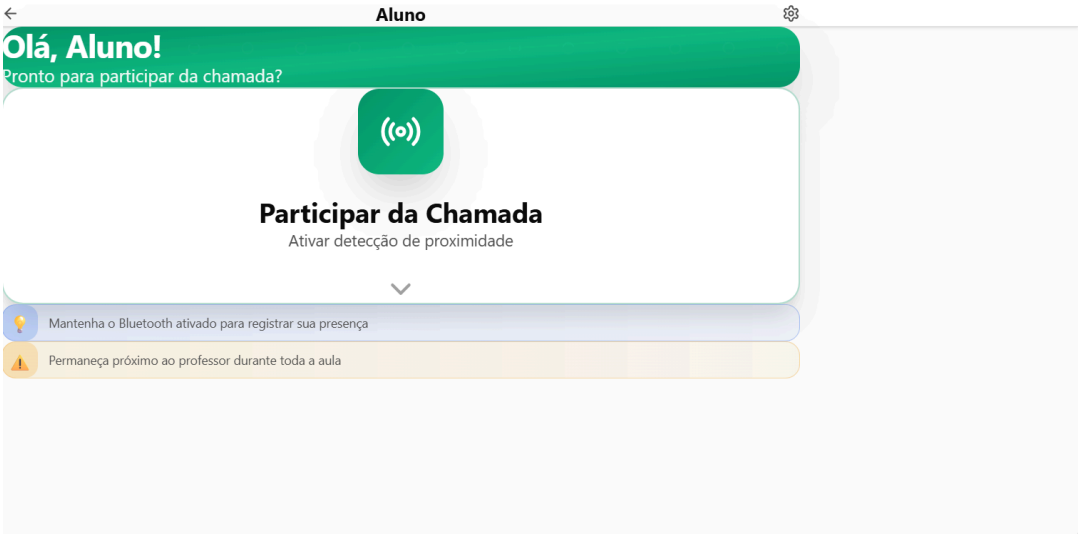
Protótipos de tela

1 - Tela Inicial

2 - Menu Professor

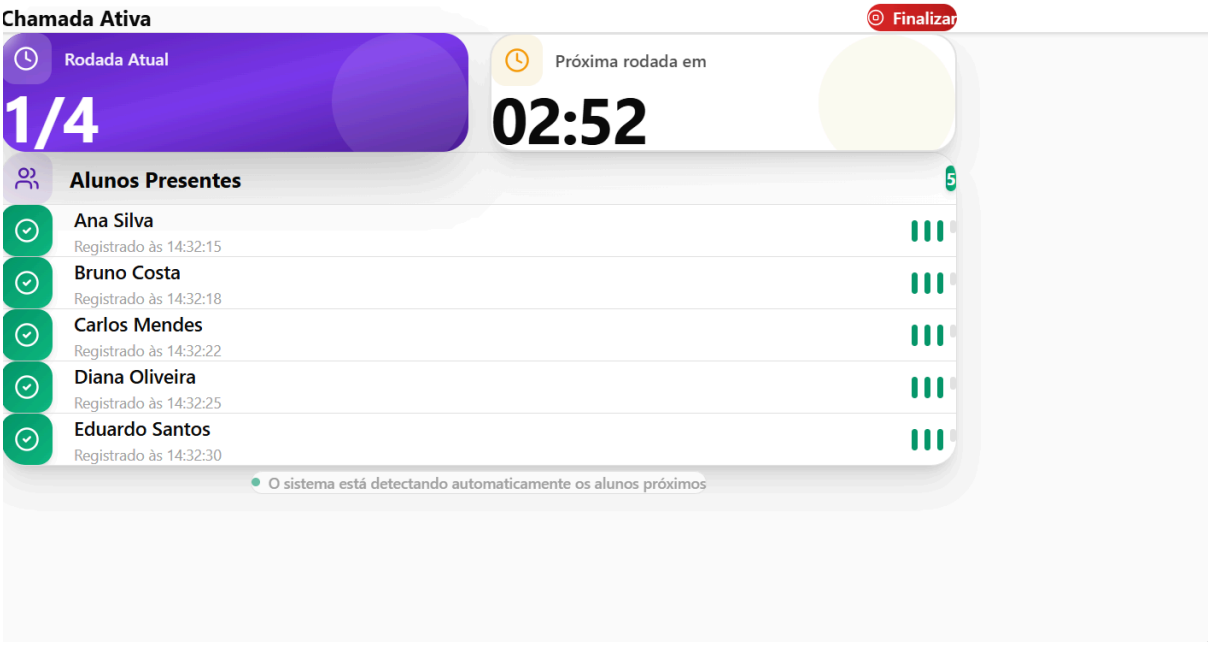


3 - Menu Aluno

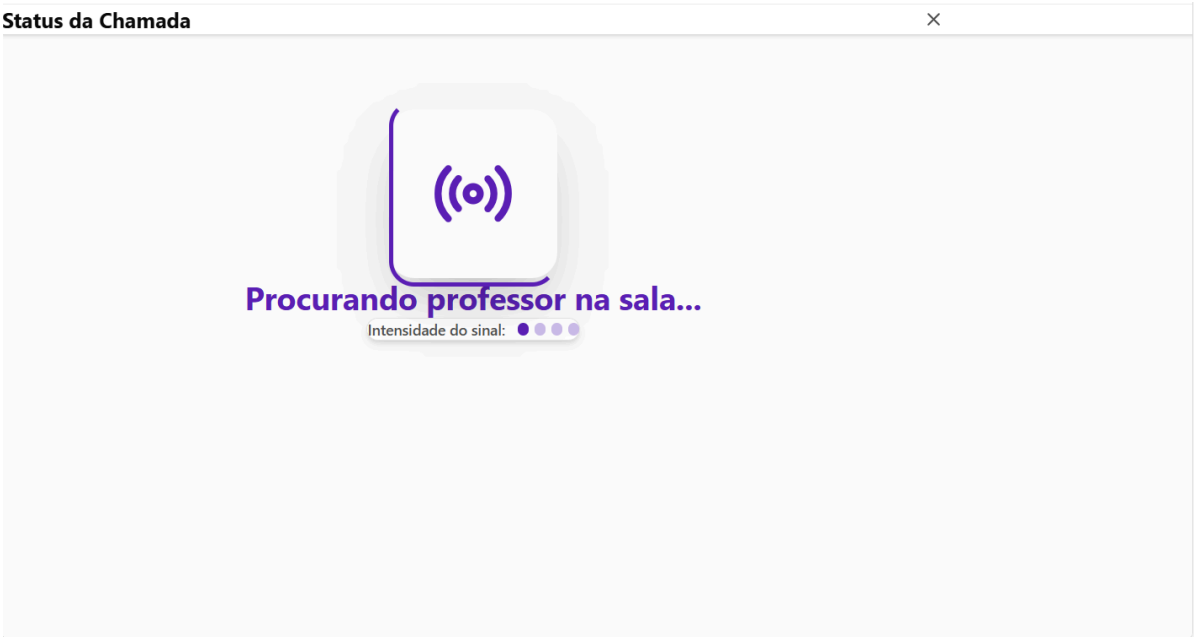


4 - Relatório de Chamada

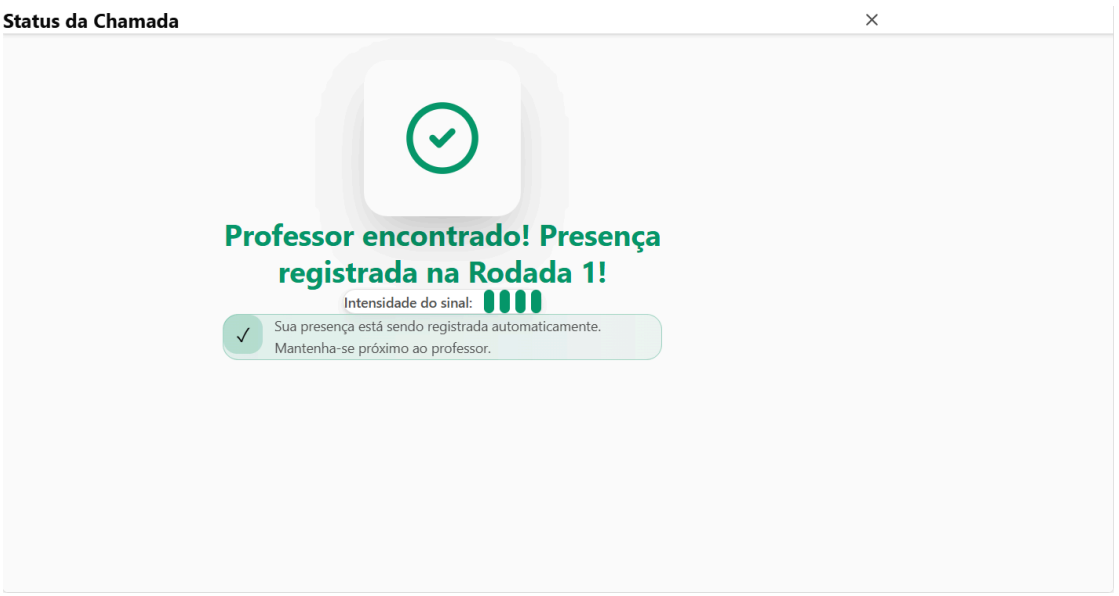
5 - Chamada ativa



6 - Chamada em andamento



7 - Chamada efetuada


Status da Chamada
Professor encontrado! Presença registrada na Rodada 1!
Intensidade do sinal:
Sua presença está sendo registrada automaticamente.
Mantenha-se próximo ao professor.