## ATIVIDADES CHAVE

### Priorização Eficiente:

Crucial para garantir que chamadas de emergência sejam atendidas com a urgência apropriada. Essa atividade envolve implementar algoritmos e sistemas que classificam e priorizam as chamadas com base na gravidade e na urgência, garantindo que os casos mais críticos recebam atenção imediata.

A ordem de prioridade é:

1. Chamada automática
2. Chamada de emergência
3. Chamada de auxilio

### Chamada Automática:

Chamada automática refere-se à capacidade de o sistema realizar chamadas automaticamente em resposta a uma emergência ou para comunicar-se com serviços de emergência. Isso exige o desenvolvimento e a implementação de tecnologia de automação de chamadas, incluindo integração com sistemas de comunicação e protocolos de resposta rápida.

A chamada automática será ativada se algum desses critérios for atendido:

* Frequência Cardíaca (Pulso) em Bradicardia, abaixo de 60 bpm
* Frequência Cardíaca (Pulso) em Taquicardia, acima de 100 bpm
* Frequência Respiratória em Bradipneia, abaixo de 12 respirações por minuto
* Frequência Respiratória em Taquipneia, acima de 20 respirações por minuto
* Pressão Arterial em Hipotensão, abaixo de 90/60 mmHg
* Pressão Arterial em Hipertensão, acima de 180/120 mmHg
* Saturação de Oxigênio (SpO2) em Hipoxemia, abaixo de 90%

### Feedback Visual Imediato:

É essencial para garantir que os usuários do sistema saibam imediatamente que sua chamada foi recebida e está sendo processada. Isso exige o desenvolvimento de interfaces visuais que forneçam confirmações e informações em tempo real.

### Acesso ao Prontuário:

Permite que os usuários acessem os prontuários médicos de seus pacientes por meio de uma tela próxima ao leito. Ao apresentar o seu id (qrcode ou barcode) no leitor que vai estar próximo do leito, se estiver liberado no sistema a visualização do prontuário para o mesmo, a ficha médica do paciente em questão irá aparecer no display.

Esse sistema permite que os enfermeiros(as) em uma situação de emergência consigam acessar rapidamente os dados médicos do paciente e protege de pessoas que não podem visualizar as informações ali presentes.

## Recursos Chave

### Node-RED:

Node-RED é uma plataforma de programação visual que facilita a conexão de dispositivos, APIs e serviços online de maneira simples e intuitiva. Escolhemos utilizar essa ferramenta por sua facilidade de uso e implementação, tanto com o Raspberry Pi quanto com o ESP32.

### Node.js:

Node.js é um software de código aberto, multiplataforma, baseado no interpretador V8 do Google e que permite a execução de códigos JavaScript fora de um navegador web. Por fornecer uma boa performance, vamos adotar o Node.js como uma solução viável e eficaz de tecnologia para Back-end, como o próprio Google, Netflix, entre outros.

### React.js:

React.js é uma biblioteca JavaScript criada pelo Facebook para desenvolver interfaces de usuário (UI) dinâmicas e responsivas em aplicações web. É amplamente utilizada em projetos que exigem UIs ricas e interativas. Optamos por usar o React.js por simplificar e facilitar a construção de interfaces intuitivas para o usuário.

### Socket.io:

Socket.io é uma biblioteca que permite comunicação bidirecional em tempo real entre clientes e servidores. Ideal para atualizações instantâneas e notificações em aplicações web, como dashboards e jogos online. Escolhemos essa ferramenta porque facilita o desenvolvimento de aplicações com comunicação instantânea, como chats e atualizações em tempo real, devido à sua baixa latência.

### MySQL:

O MySQL é o banco de dados de código aberto mais conhecido no mundo. De acordo com o DB-Engines, o MySQL é o segundo banco de dados mais popular, ficando atrás do Oracle Database. Optamos pelo MySQL por sua rapidez, confiabilidade e escalabilidade. Ele é fácil de usar e instalar, e tem sido testado em diversos cenários. Com desempenho superior e alta disponibilidade, o MySQL atende nossas demandas mais exigentes, suportando bilhões de usuários e oferecendo segurança robusta.