UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
ESCOLA DE CIÊNCIAS E TECNOLOGIA
CURSO DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

COMPONENTE CURRICULAR: LÓGICA DE PROGRAMAÇÃO

DOCUMENTAÇÃO DO NEUROAPP: UMA SOLUÇÃO MÓVEL INTEGRADA PARA CONTROLE E TELEMETRIA DE ROBÔ AUTÔNOMO

RESUMO

Este artigo apresenta o início do desenvolvimento e a arquitetura de um aplicativo móvel multiplataforma, projetado como interface principal para controle e monitoramento de um robô autônomo baseado em microcontrolador ESP32. O projeto tem como objetivo oferecer uma ferramenta robusta para desenvolvedores e operadores realizarem testes, validação de hardware e firmware, além de acompanhamento de telemetria em tempo real.

A aplicação está sendo desenvolvida em React Native, garantindo compatibilidade com sistemas Android e iOS, e utiliza o Bluetooth Low Energy (BLE) como principal meio de comunicação local. O sistema adota ainda uma estratégia de persistência de dados híbrida, combinando armazenamento local e sincronização em nuvem através da plataforma própria da NeuroBeep, assegurando o funcionamento offline e a atualização automática quando a conexão é restabelecida.

1. Arquitetura e Conectividade

A arquitetura do aplicativo foi concebida com foco em conectividade, resiliência e modularidade. A comunicação entre o dispositivo móvel e o robô é estabelecida via Bluetooth Low Energy (BLE), permitindo o envio e recebimento de comandos de controle e dados de sensores.

O sistema está sendo planejado para operar tanto online quanto offline. No modo offline, os dados são armazenados localmente utilizando WatermelonDB. Já em modo online, ocorre a sincronização automática com o banco de dados em nuvem da NeuroBeep, garantindo a integridade e a persistência das informações coletadas.

2. Painel de Controle (Dashboard)

O módulo principal da aplicação, DashBoard.js, concentra as funcionalidades de monitoramento do robô, exibindo informações de telemetria e status de conexão. A arquitetura desta tela é organizada em componentes independentes, o que permite fácil expansão e manutenção. Entre os recursos implementados, destacam-se:

- Conectividade BLE: gerenciamento do estado de conexão com o robô, incluindo comandos de conexão e desconexão.
- Comandos de Sistema: Execução de ações como *Start*, *Reset Kalman*, *Calibrar IMU* e *Testar*, destinadas a testes e validação de firmware.

- Marcadores: controle e visualização de parâmetros como *Contador* e *Distância percorrida*, com ajuste dinâmico de espaçamento.
- Telemetria: exibição de dados em tempo real provenientes de sensores — encoders, velocidades, odometria e coordenadas GPS — com ajuste da taxa de atualização em hertz.
- Logs: registro e exibição de mensagens do sistema, permitindo depuração e análise de eventos.

A organização modular permite que cada funcionalidade seja tratada como um "cartão" independente, atendendo diretamente aos requisitos funcionais

3. Controle Manual (Control Screen)

O módulo ControlScreen.js implementa a interface de controle remoto manual do robô, atendendo ao requisito. A tela adota uma orientação horizontal fixa para melhor ergonomia e apresenta dois joysticks virtuais desenvolvidos com react-native-gesture-handler e react-native-reanimated. O joystick esquerdo controla a velocidade linear (movimentos para frente e para trás), enquanto o direito controla a rotação angular (movimentos de direção). Essa estrutura proporciona uma experiência de controle intuitiva e responsiva.

4. Conclusão e Perspectivas Futuras

O aplicativo NeuroApp busca cumprir os principais requisitos de um sistema móvel para controle e telemetria de robôs autônomos, oferecendo conectividade BLE estável, interface modular e operação híbrida (online/offline).

Como próximos passos, está prevista a integração de inteligência artificial local (on-device AI) utilizando ExecuTorch, e a implementação de um rosto robótico 3D interativo em react-three/fiber, transformando o aplicativo em um bot de conversação interativo com representação visual e respostas inteligentes — consolidando a NeuroBeep como uma plataforma de experimentação em robótica inteligente móvel.

REFERÊNCIAS:

DOTINTENT. **React Native BLE Plx**. Disponível em: <https://dotintent.github.io/react-native-ble-plx>. Acesso em: 12 nov. 2025.

META PLATFORMS, INC. **React Native Documentation**. Disponível em: <https://reactnative.dev/docs/getting-started>. Acesso em: 12 nov. 2025.

NOZBE. **WatermelonDB Documentation**. Disponível em: <https://watermelondb.dev/docs>. Acesso em: 12 nov. 2025.

POIMANDRES. **React Three Fiber Documentation**. Disponível em: <https://docs.pmnd.rs/react-three-fiber/getting-started/introduction>. Acesso em: 12 nov. 2025.

PYTORCH. **ExecuTorch Documentation**. Disponível em: <https://pytorch.org/executorch/stable/>. Acesso em: 12 nov. 2025.

SOFTWARE MANSION. **React Native Gesture Handler Documentation**. Disponível em: <https://docs.swmansion.com/react-native-gesture-handler/docs/>. Acesso em: 12 nov. 2025.

SOFTWARE MANSION. **React Native Reanimated Documentation**. Disponível em: <https://docs.swmansion.com/react-native-reanimated/docs/>. Acesso em: 12 nov. 2025.