

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS

**Sistema de Controle Sem Fio de Baixo Custo para
Verificação de Toques Para Esgrima Olímpica**

Relatório Final - POC II

MARCO ANTÔNIO SILVEIRA SOUZA ALVES

Orientador: Marcos Augusto dos Santos

Belo Horizonte

Setembro, 2025

1 Introdução

A esgrima olímpica é um esporte de combate que combina técnica, velocidade e precisão, praticado hoje com o apoio de aparelhos eletrônicos responsáveis por registrar os toques válidos, marcar a pontuação e controlar o tempo de combate. Esses aparelhos, porém, ainda são majoritariamente cabeados: fios conectam as armas ao aparelho central e à estrutura da pista, o que torna a montagem trabalhosa, exige um espaço dedicado exclusivamente à prática e encarece a implantação de novas pistas em clubes e escolas.

Recentemente surgiram algumas soluções comerciais de pontuação sem fio para a esgrima, buscando maior mobilidade. Apesar dos avanços, a adoção desses sistemas ainda é limitada por dois fatores principais, confiabilidade e preço. Por se tratar de um esporte olímpico, a esgrima é rigidamente regulamentada pela Federação Internacional de Esgrima (FIE), que define limites de tempo e sincronização muito restritos, como a janela de 40 ms para detecção de toques duplos na espada. Qualquer atraso ou perda de mensagem pode resultar em decisões erradas de pontuação. Além disso, os poucos aparelhos sem fio considerados confiáveis tem um custo muito alto para a realidade brasileira.

No projeto anterior foi desenvolvido um protótipo sem fio de baixo custo baseado em microcontroladores ESP32, com comunicação via ESP-NOW e implementação em MicroPython. O sistema utilizava mensagens curtas, sincronização de relógio entre servidor e clientes e confirmações em nível de aplicação para registrar toques simples e duplos respeitando o limite de 40 ms estabelecido pela FIE. Embora os resultados de latência e precisão tenham sido promissores, a avaliação concentrou-se em cenários controlados, com simulação de toques e sem integração a espadas reais ou a uma infraestrutura completa de combate, como placar visível, cronometragem ou sinalização audiovisual.

Este projeto tem como objetivo aproximar esse protótipo experimental ao formato de um aparelho de pontuação utilizável em treinos. Para isso, o sistema foi reorganizado em torno de uma máquina de estados finitos que modela o fluxo do combate, desde a inicialização e sincronização dos dispositivos até a análise dos toques e atualização da pontuação. Foram integrados marcadores audiovisuais (LEDs RGB, matriz de LED para placar e sinal sonoro), mecanismos de cronometragem, além de um estado que permitem aos próprios atletas ajustarem o estado do combate quando necessário, removendo toques inválidos e controlando o estado de pausa, efetivamente transformando as espadas em controles. Por fim, o protótipo foi conectado a espadas reais e avaliado em situações práticas de treino, de forma a validar não apenas o desempenho de rede mas também a usabilidade e a robustez do sistema em um ambiente mais próximo da realidade dos clubes de esgrima.

2 Referencial Teórico

Podemos observar alguns trabalhos acadêmicos voltados para o desenvolvimento de sistemas de pontuação sem fio para a esgrima, como em Laurenson and Redoute [2012], Joseph Anthony Alvarez [2008]. De forma geral, essas propostas exploram o uso de rádios de baixa potência acoplados às armas ou aos coletes, com o objetivo de eliminar os cabos e tornar a montagem das pistas mais flexível. No entanto, a maior parte dos estudos foca em outras armas da esgrima olímpica, como sabre e florete, e nem sempre discutem em detalhe os requisitos mais restritos de temporização e confiabilidade necessários, como o registro correto de toques duplos em uma janela de 40 ms.

No mercado encontramos alguns aparelhos de controle sem fio, como os apresentados em Calibur Team [2025], Hitmate Fencing Ltd [2025], EnPointe Fencing Systems [2025], que prometem resolver a questão da mobilidade e facilitar a montagem de áreas de treino e competição. Em tese, esses sistemas permitem que clubes de esgrima montem pistas em espaços compartilhados com outros esportes ou até em locais improvisados, sem a necessidade de fixação permanente de cabos no chão ou no teto. Além disso, alguns desses equipamentos trazem integração com aplicativos para celular e soluções de streaming, ajudando na transmissão de combates e na visualização dos resultados em tempo real. Porém, relatos de usuários apontam problemas recorrentes de confiabilidade na detecção de toques, perda de sincronia entre as unidades, necessidade frequente de manutenção das baterias e falhas intermitentes no registro da pontuação. Somando-se a isso, os modelos acabam custando valores muito altos, o que torna a adoção em clubes pequenos, projetos sociais e escolas praticamente inviável.

Olhando para outros esportes de combate, também existem propostas de sistemas de pontuação sem fio em modalidades como o Kendo [Elzinga, 2015] e o Taekwondo [Partridge et al., 2005]. Em geral, essas soluções investem bastante na detecção de golpes por meio de sensores de impacto, acelerômetros ou sensores de pressão integrados ao uniforme ou aos equipamentos de proteção. Apesar de trazerem ideias interessantes para a captura automática de eventos muitas vezes esses sistemas partem de uma premissa diferente da adotada neste trabalho: elas pressupõem alterações físicas nos equipamentos oficiais do esporte ou o uso de roupas especiais, o que dificultaria a adoção em contextos mais simples de treino e inclusive iria contra os regulamentos da FIE. Além disso, a discussão sobre a confiabilidade da comunicação sem fio, a sincronização dos eventos e o impacto da latência na decisão de pontuação costuma aparecer de maneira mais superficial, sem o mesmo nível de detalhamento necessário para um sistema de arbitragem eletrônica.

No contexto de dispositivos embarcados, alguns trabalhos analisam o uso do ESP32

em aplicações de Internet das Coisas (IoT) e comparam diferentes formas de programação, como Arduino Core, ESP-IDF e MicroPython [Babiuch et al., 2019]. Os autores destacam que o MicroPython se mostra bastante atrativo para prototipagem e para projetos em que a legibilidade do código e a facilidade de manutenção têm peso importante, o que se encaixa bem no nosso cenário de um aparelho de baixo custo voltado para clubes e projetos sociais. No nosso caso o uso de MicroPython também facilita a organização em módulos e a implementação de uma máquina de estados bem definida, aumentando a manutenibilidade do sistema.

Há ainda estudos que focam na segurança de sistemas IoT, como em Ciklabakkal et al. [2019]. Embora a segurança não seja o foco central deste projeto, esses estudos reforçam a importância de pensar desde cedo em mecanismos básicos de autenticação e proteção contra mensagens falsas, especialmente em um contexto de competição ou em ambientes com muitas redes sem fio compartilhando o mesmo espaço físico.

Por fim, uma parte relevante da literatura e da documentação técnica discute o protocolo ESP-NOW, utilizado neste projeto como base da comunicação entre as placas [Systems, 2024, UI, 2020]. Um dos principais pontos fortes do ESP-NOW é sua baixa latência, e nossos testes mostram que ele é capaz de transmitir pacotes em poucos milissegundos mesmo com criptografia e confirmação de entrega ativadas, ficando confortavelmente abaixo do limite de 40 ms definido pela FIE para a detecção de toques simultâneos. Além disso, o protocolo funciona bem em ambientes abertos, com vários dispositivos comunicando-se ao mesmo tempo, o que é interessante para salas de treino com múltiplas pistas. A literatura e relatos de comunidade indicam, porém, que o uso do ESP32 em aplicações alimentadas por bateria exige atenção ao dimensionamento da fonte e ao impacto dos picos de consumo na estabilidade do sistema [ESP32 Community, 2024, Forum, 2020]. Esses pontos abordam diretamente os desafios enfrentados neste trabalho, que busca um equilíbrio entre simplicidade, confiabilidade e baixo custo em um cenário de uso real.

3 Metodologia

A metodologia que foi utilizada para o desenvolvimento desse trabalho é definida a seguir.

- **Implementação da máquina de estados:** Tendo como base a arquitetura atual do protótipo foram feitas melhorias no fluxo do protótipo anterior e foi implementada uma máquina de estados que modela o sistema, facilitando alterações no comportamento do sistema e na inclusão de novos fluxos/estados.
- **Marcadores audiovisuais:** Implementada uma interface de controle audiovisual para gerenciar os leds, placares e *buzzers* presentes no sistema, em conjunto com

- **Sistema de cronometragem:** Nesta etapa, já com a máquina de estados consolidada, implementamos o módulo de cronometragem de tempo e estendemos a máquina de estados para inclusão de um novo fluxo: A pausa. Neste estado as espadas dos esgrimistas assumem um papel de controles, possibilitando que os atletas façam alterações no sistema sem necessidade de um terceiro.
- **Montagem do protótipo:** Neste ponto a ferramenta estava pronta em nível de código, e foi feita a montagem física do modelo completo. Envolveu o estudo do circuito para alimentação das ESP32 por baterias, a soldagem dos componentes em placas de fenolite e a montagem final.
- **Análise dos resultados e correções de falhas:** Realizados testes reais e correções de bugs encontrados na maquina de estados, nos algoritmos de controle de temporização e/ou autenticação ou na montagem do hardware como problemas nas soldas, cabos ou periféricos.

4 Arquitetura do Sistema

A arquitetura do sistema proposto segue a mesma linha do trabalho anterior, mantendo a ideia de um servidor central responsável pela lógica de arbitragem e dois clientes ligados às armas dos atletas. Cada nó é implementado com um microcontrolador ESP32, comunicação sem fio via ESP-NOW e um conjunto de periféricos de baixo custo, suficientes para uso em treinos e em projetos sociais de iniciação à esgrima.

Neste POC II, além de refinar o código e resolver *bugs* nos algoritmos de sincronização e inicialização, o foco passou a incluir a experiência de uso em treinos reais. Foram integrados marcadores visuais com LEDs endereçáveis, um placar com cronômetro em matriz de LEDs, um sinal sonoro simples e um esquema de alimentação por bateria pensado para ambientes sem infraestrutura elétrica dedicada. A lógica de controle foi organizada em torno de uma máquina de estados finitos (MSF) compartilhada entre servidor e clientes, o que facilita a manutenção, leitura e expansão do código.

4.1 Hardware

O sistema é composto por três nós principais: um nó *servidor*, que concentra a lógica de arbitragem e o controle do placar, e dois nós *clientes*, conectados às armas dos esgrimistas. Todos são baseados em módulos ESP32, mais especificamente ESP32-C6, escolhidos por combinarem conectividade sem fio, capacidade de processamento e baixo custo [Babiuch et al., 2019].

Nos nós clientes, o ESP32 é ligado ao circuito da espada que possui um pino de toque capacitivo na sua ponta e é configurado a um GPIO qualquer com lógica de *pull-down* e detecção de borda de subida. Essa entrada detecta o fechamento do circuito de toque, de forma análogo aos aparelhos tradicionais mas sem a necessidade de cabo até o aparelho central. A mesma entrada é utilizada para gestos de controle pelo próprio atleta: o código implementa detecção de toques curtos e toques longos (na escala de segundos), permitindo que comandos como pausa ou remoção de um ponto inválido sejam feitos a partir da própria arma, sem depender de botões físicos adicionais, e o que diferencia se o toque é um ponto ou um comando de controle acaba sendo o estado em que a MSF se encontra. Além disso os clientes contam com uma solução de alimentação por bateria, com célula 18650, módulo de carga TP4056, regulador LDO TLV75533 e uma chave KCD11 para ligar e desligar o conjunto. Tais componentes são frequentemente utilizados em conjunto com microcontroladores para montagem de circuitos com alimentação por baterias *Li-Íon* [Santos and Santos, 2023, Espressif Systems, 2022]

O nó servidor agrega mais periféricos, pois é o responsável pela interface com os usuários que estão fora da pista. Ele reúne um ESP32 para a lógica de controle, duas matrizes 4x4 de LEDs RGBs WCMCU-2812B-16 um para cada lado da pista (verde e vermelho), uma matriz de LEDs formada por quatro módulos 8x8 com driver MAX7219 em cascata, que exibem placar e cronômetro, um buzzer passivo controlado por PWM para o sinal sonoro de toque e fim de tempo.

Do ponto de vista mecânico, o protótipo foi montado inicialmente em *breadboards* e fios tipo *jumper*, de forma a facilitar a experimentação em laboratório e a troca de componentes durante o desenvolvimento. Uma vez que as alterações de código foram finalizadas, as ESP32 foram soldadas em placas de fenolite com soldas de estanho e cabos 26AWG.

4.2 Comunicação

Assim como no POC I, a comunicação entre os nós é feita sem fio com o protocolo ESP-NOW, que permite o envio de pequenas mensagens entre ESP32 sem necessidade de um roteador Wi-Fi ou de associação a uma rede tradicional [Systems, 2024, UI, 2020]. No início da execução, servidor e clientes se adicionam mutuamente como *peers*, usando os endereços MAC pré-configurados. Uma função de apoio centraliza essa configuração, simplificando o código das máquinas de estados.

As mensagens trocadas são curtas e baseadas em texto, com formatos bem definidos, como `SYNC_US:<timestamp>` para sincronização, `TOUCH:<timestamp>` para sinalizar um toque detectado e `TOUCH_RECEIVED` para confirmação de recebimento por parte do servidor,

e todas as mensagens são criptografadas usando HMAC antes de serem enviadas.

Do ponto de vista da arquitetura, o servidor atua como ponto central: recebe os eventos de toque dos dois clientes, decide o resultado, atualiza seus LEDs, placar e cronômetro e entra em pausa aguardando comandos dos clientes. A lógica detalhada de análise temporal entre os toques e de detecção de toques duplos é discutida em seção posterior.

4.3 Máquina de estados e controle de pausa

Para organizar o fluxo do combate e separar melhor as responsabilidades entre servidor e clientes, o sistema foi estruturado em torno de uma MSF. O sistema define uma classe genérica **StateMachine**, que mantém o estado atual, a tabela de transições e *handlers* de entrada e saída para cada estado. Sobre essa base são construídas duas máquinas específicas: uma para o servidor (**ServerStateMachine**) e outra para os clientes (**ClientStateMachine**).

No servidor, após a fase de **SYNC**, a máquina entra no estado **PAUSED** e permanece assim com ambos os LEDs em branco até receber um comando de retomada enviado pelos clientes.

Nos clientes, quando em **PAUSED**, não é registrado toques de combate, mas é monitorado continuamente o GPIO de entrada para toques de comando vindos dos esgrimistas. Toques curtos são interpretados como um pedido de retomada, e o cliente gera uma mensagem **RESUME:<timestamp>**, envia essa mensagem assinada ao servidor e, em caso de sucesso, dispara o evento **EVENT_RESUME** localmente. Quando o servidor decide que ambos os clientes solicitaram a retomada, ele envia **START_MATCH** para os dois, e as máquinas de estados dos clientes passam para **WAITING_TOUCH**, habilitando a detecção de toques de fato. Ainda no estado de pausa temos o comando de correção de pontuação. Um toque longo na arma é interpretado como um pedido explícito de subtração de ponto. Nesse caso, o cliente não dispara **EVENT_RESUME**, mas constrói uma mensagem **SUBTRACT** e a envia ao servidor. Ao receber essa mensagem, o servidor identifica de qual cliente ela veio, associa esse cliente a uma das cores (verde/esquerda ou vermelho/direita) e, se o placar for maior que zero, decrementa em um ponto a pontuação correspondente. Em seguida, o placar é atualizado na matriz de LEDs e o buzzer emite dois bipes curtos para indicar que a correção foi aplicada, sem tirar o sistema do estado **PAUSED**.

Na prática, essa combinação de máquina de estados e gestos na própria espada permite que os atletas controlem o fluxo do treino sem precisar acessar botões físicos no aparelho ou um agente terceiro, comumente um árbitro. Isso possibilita o uso de cronômetros e placares por esgrimistas em situação de treino, algo pouco usual pois os

aparelhos atualmente que possuem estes marcadores visuais dependem de um controle remoto externo e geralmente são utilizados apenas em competições. Como a presença de arbitragem é geralmente inexistente durante treinos, a utilização da espada como um controle acaba sendo necessário para o correto funcionamento de tais componentes extras.

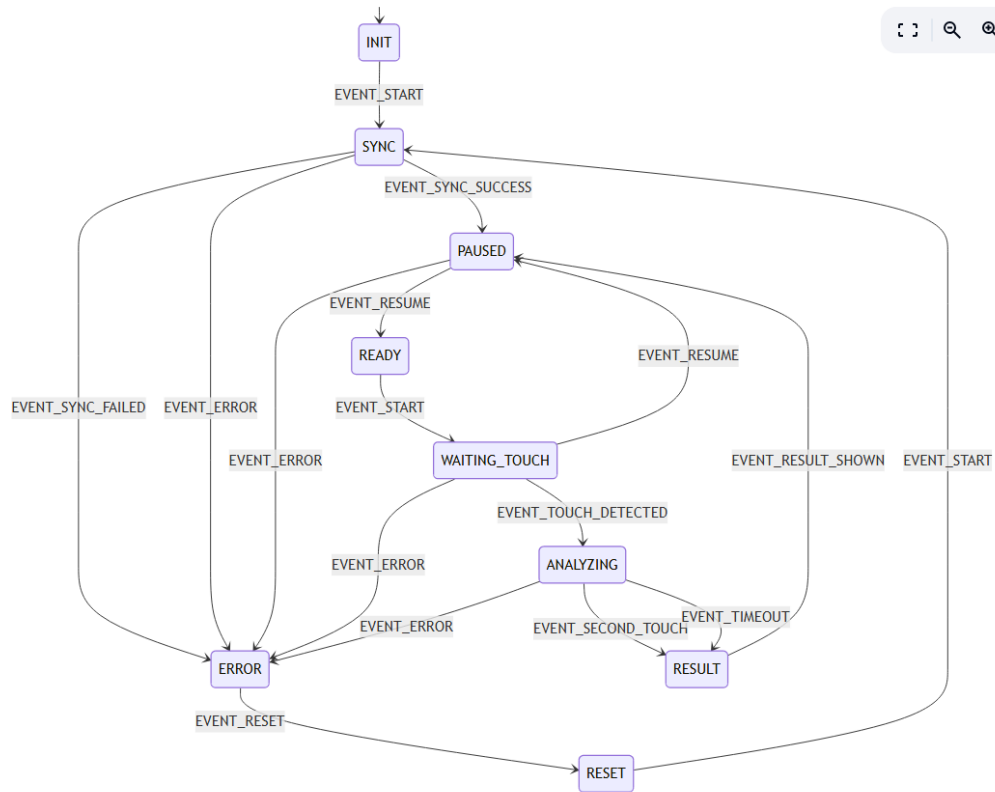


Figura 1 – Visão geral da máquina de estados e suas transições.

4.4 Marcadores visuais com LEDs endereçáveis

Os marcadores visuais de cada lado da pista são matrizes de LEDs endereçáveis WCMCU-2812B-16 (16 LEDs WS2812B organizados em um quadrado 4x4) [Worldsemi, 2020] controlados por objetos definidos no código em um módulo controlador, que encapsula as operações básicas de ligar, desligar, preencher com uma cor uniforme e realizar *piscas* temporizados. Tais controles são feitos a partir da biblioteca *neopixel* como visto em Adafruit Industries [2024], já disponível nativamente em distribuições do MicroPython por meio de MicroPython Project [2024a]

O servidor utiliza os LEDs para indicar de forma clara o estado do combate. Cores diferentes são usadas para cada situação: ambos os LEDs brancos quando o sistema está em *pause*, apagado quando os atletas estão prontos, cores distintas para cada lado em caso de toque válido e padrões de pisca para sinalizar eventos especiais como erros, reinício da sincronização ou fim do tempo.

4.5 Placar e cronômetro em matriz de LEDs

O placar e o cronômetro foram implementados em uma matriz de LEDs 32×8 formada por quatro módulos 8×8 com driver MAX7219 [Maxim Integrated, 2019, Adafruit Learning System, 2024] em cascata. Uma classe `LedMatrixScoreboard`, que organiza o layout em três regiões, foi desenvolvida de tal forma que as oito primeiras colunas exibem o placar do atleta da esquerda com um dígito grande, as 16 colunas centrais exibem o tempo no formato *M:SS* com uma fonte menor e as oito colunas finais exibem o placar do atleta da direita, também com dígito grande.

Por questões de *pinout* do ESP32 utilizado, a comunicação com os módulos MAX7219 foi feita via *SoftSPI* [MicroPython Project, 2024b], em vez de usar diretamente o barramento por hardware. A biblioteca não está nativamente disponível na versão do MicroPython instalada, porém foi importada manualmente a partir do projeto mcauser [2017].

O controle do tempo é feito em segundos, o que simplifica a integração com a máquina de estados do servidor. Eventos como início de combate, pausa, retomada e fim de tempo apenas ajustam a variável responsável por contar o tempo, e a classe do placar se encarrega de converter esse valor para o formato visual adequado.

4.6 Alimentação e baterias

O sistema foi pensado desde o início para operar com baterias recarregáveis. Nos nós clientes, a alimentação é feita a partir de uma célula de *Li-Ion* ICR18650 26J, com capacidade nominal de 2600 mAh e tensão nominal de 4,15 V e o carregamento é realizado por um módulo TP4056 [NanJing Top Power ASIC Corp., 2019].

A saída da bateria é estabilizada em 3,3 V por um regulador de baixa queda TLV75533 [Texas Instruments, 2022], seguindo as recomendações de projeto de hardware da Espressif para o ESP32 [Espressif Systems, 2022]. O microcontrolador são alimentados por essa linha de 3,3 V. Uma chave KCD11 no caminho principal da bateria permite desligar completamente o nó quando não está em uso.

Durante os testes, foi observado que a combinação de bateria 18650 com módulo carregador TP4056 e LDO regulador TLV75533 é suficiente para várias horas de treino com consumo moderado de LEDs, desde que o brilho dos WS2812B seja configurado para valores intermediários, e a utilização de cabos AWG 26 ajuda a manter o conjunto seguro. A literatura em conjunto com relatos de comunidade indicam, porém, que o uso do ESP32 em aplicações alimentadas por bateria exige atenção ao dimensionamento da fonte e ao impacto dos picos de consumo na estabilidade do sistema ESP32 Community

[2024], Forum [2020], algo que foi levado em conta no dimensionamento do protótipo.

Por enquanto, o sistema não implementa monitoramento detalhado do nível de bateria. Essa proteção fica a cargo dos módulos de carga/proteção e das próprias características de desligamento do ESP32. Mesmo assim, a adoção de componentes comuns como carregadores USB-C para alimentar o TP4056 ajuda a tornar o protótipo mais próximo da realidade de clubes e projetos sociais.

5 Resultados

Nesta seção apresentamos os principais resultados obtidos com o POC II, com foco no funcionamento geral do sistema, na comunicação sem fio, na experiência de uso em treinos e no comportamento do protótipo quando alimentado por baterias. Enquanto o POC I tinha um caráter mais experimental, voltado a medir latências e validar ideias em cenário controlado, o POC II buscou responder se era realmente viável produzir um protótipo funcional e aplicá-lo a um treino de esgrima.

E ao final do projeto foi possível produzir um protótipo funcional, de baixo custo, que respeita os requisitos básicos das normas técnicas da FIE [Federation, a,b] no que diz respeito à detecção de toques e ao uso de janelas temporais bem definidas, ainda que o foco esteja em treinos e projetos sociais, e não em competições oficiais.

Em testes com espadas reais, o sistema se mostrou capaz de acompanhar um combate sem problemas, registrando com precisão os toques simples e duplos, e teve uma excelente aceitação. Nos testes, o controle de pausa e correção de pontuação esquema se mostrou intuitivo e reduziu a necessidade de interação com botões físicos no aparelho.

Erros de arbitragem comuns em treino ou toques inválidos como toques ao chão puderam ser corrigidos rapidamente com o mecanismo de controle a partir da lâmina, que apesar de não tem paralelo direto nos aparelhos tradicionais, se encaixa bem na proposta de um equipamento voltado para treinos e projetos sociais, entregando autonomia aos atletas e uma redução da complexidade da interface física.

Na prática, isso significa que o conjunto duas espadas com servidor e placar se comporta como um aparelho de treino utilizável: é possível montar a pista, ligar os três nós, sincronizar o sistema e realizar séries de combates curtos sem precisar ajustar configurações internas ou reiniciar o protótipo com frequência. Ao longo dos testes, não foram observados *resets* aleatórios do ESP32 nem travamentos recorrentes da máquina de estados enquanto o servidor estava alimentado diretamente na tomada, o que indica que a solução de alimentação e a lógica de controle estão suficientemente estáveis para o uso pretendido. Contudo, quando conectado em uma bateria externa, o servidor apresentou algumas instabilidades e *resets* com mensagens de erro indicando *brownout* [Microchip Technology Inc., 2018], ou seja, uma queda na tensão de alimentação suficiente para colocar o ESP32

em um estado instável, disparando erros na MSF. Isso indica uma possível ineficiência energética do circuito do servidor, sendo necessária uma revisão futura.

Em resumo, o POC II atingiu os objetivos traçados: foi construído um protótipo sem fio, de baixo custo, alimentado por baterias, capaz de registrar toques de espada em tempo hábil, exibir o resultado em um placar visível e permitir que os próprios atletas controlem pausas e correções de pontuação a partir da lâmina. As medições de latência do POC I e os testes práticos indicam que, mesmo com os *overheads* adicionais do ESP-NOW estendido, da assinatura das mensagens via HMAC e do uso de MicroPython, ainda há folga temporal suficiente para evoluir o sistema na direção de janelas de análise mais restritivas, caso se deseje aproximá-lo de um aparelho de competição no futuro.

Ao mesmo tempo, a modularidade do código e a simplicidade dos componentes utilizados deixam o caminho aberto para adaptações: desde versões mais enxutas, com menos LEDs e sem placar, até configurações mais completas, integradas a aplicações web ou a sistemas de transmissão de vídeo para clubes e projetos sociais de esgrima.

6 Conclusão e Trabalhos Futuros

6.1 Considerações Gerais

Sendo apenas um protótipo, o modelo possui espaço para muitas melhorias tanto do ponto de vista de software quanto de hardware, mas se apresenta de maneira muito promissora. Hoje já temos um sistema completamente funcional e suficiente para treinos nas salas de esgrima, de fácil produção e replicação, e que pode servir como ferramenta para expansão do esporte e sua popularização e democratização por meio de projetos sociais. Aparelhos sem fio mais baratos ficam na faixa de R\$ 5.300 em conversão direta, com aparelhos mais caros custando até exorbitantes R\$ 10.000, enquanto o protótipo aqui apresentado possui custo aproximado de R\$ 250, além de características não presentes nos modelos comerciais, como o placar, cronômetro e sistema de controle de estados pelos próprios esgrimistas, eliminando em muitos contextos a necessidade de um árbitro dedicado para treinos.

Do ponto de vista técnico, o POC II mostrou que é viável construir um sistema sem fio de esgrima baseado em ESP32, ESP-NOW e MicroPython, capaz de operar com baterias, manter a sincronização entre os nós e oferecer uma interface visual clara por meio de matrizes de LEDs e indicadores RGB. A organização em módulos e em máquina de estados deixou o código mais legível e facilitou a inclusão de funcionalidades adicionais, como o controle de pausa e subtração de pontos por gestos na própria espada.

Do ponto de vista de uso, o sistema já cumpre o papel de aparelho de treino,

permitindo montar pistas em espaços improvisados, reduzir a dependência de infraestrutura fixa e tornar mais acessível a prática de esgrima em escolas, clubes pequenos e projetos sociais. Em fases futuras, é possível imaginar integrações adicionais, como um painel web para acompanhamento remoto de lutas, registro automático de resultados ou até ferramentas simples de análise de desempenho dos atletas.

Além disso é importante ressaltar que durante o desenvolvimento deste projeto e a redação deste relatório foram utilizadas ferramentas de Inteligência Artificial baseadas em LLMs para apoiar tarefas de revisão de texto, organização de seções, validação das escolhas e compatibilidades dos componentes de hardware, além de servir como ferramenta essencial para *debug* e entendimento de códigos de erro genéricos das ESP32. Esses sistemas se mostraram úteis como assistentes, ajudando a acelerar a prototipagem de ideias e a padronizar a escrita.

6.2 Trabalhos Futuros

Na questão do software, uma evolução natural é o refinamento do algoritmo de detecção de toques duplos, explorando técnicas de filtragem temporal mais robustas, junto com a implementação de monitoramento de bateria e telemetria dos nós, permitindo que o servidor exiba no placar ou em uma interface web o nível de carga de cada cliente. No módulo de telemetria também é interessante a inclusão de um modo *debug*, de tal forma que os *logs* das trocas de mensagens entre os nós bem como as transições nas MSF sejam registradas, para facilitar ainda mais o processo de validação e manutenção futuras. Além disso, a integração do sistema com um aplicativo ou painel online pode ampliar seu uso em treinos coletivos, oferecendo histórico de lutas e acompanhamento remoto.

Já no âmbito do hardware, um próximo passo importante é migrar o protótipo das montagens em placas de fenolite para uma placa de circuito impresso dedicada, aumentando a confiabilidade mecânica e facilitando a replicação do sistema em clubes ou projetos sociais. Também é possível melhorar a prototipagem das caixas que contém os módulos, de forma a ficarem mais leves e ergonômicas, facilitando seu uso no dia a dia. É importante também investigar a questão do *brownout* ocorrendo no servidor quando conectado em fontes de energia externas.

Algo não abordado neste projeto que se apresenta como um grande desafio até para os sistemas sem fio mais comerciais atualmente é a detecção de toques na guarda da espada do oponente, que devem ser considerados como toques inválidos. Por ser um problema de elétrica extremamente desafiador, este foi ignorado aqui e poderá ser revisitado em trabalhos futuros.

Referências

- Callum Sean Bain Laurenson and Jean-Michel Redoute. The design of a wireless scoring system for epee fencing. International Journal of Sports Science and Engineering, 6(4): 239 – 247, 2012. ISSN 1750-9823.
- Marvin Trevor Gamalinda Anthony Malino Joseph Miguel Joseph Anthony Alvarez, Michael Victor Cristobal. Design of a wireless scoring system for fencing using rfid technology. International Journal of Sports Science and Engineering Vol. 02 (2008) No. 02, pp. 79-85, 2008.
- Calibur Team. Calibur – Wireless Scoring for Fencing. <https://calibur.ai/>, 2025. Acessado em: 24 maio 2025.
- Hitmate Fencing Ltd. Hitmate – Wireless Scoring for Épée. <https://www.hitmate.co.uk/>, 2025. Acessado em: 24 maio 2025.
- EnPointe Fencing Systems. EnPointe – Smart Wireless Fencing System. <https://enpointefencing.com/>, 2025. Acessado em: 24 maio 2025.
- Jacob D. Elzinga. Wireless electronic scoring system for kendo competitions. Master’s thesis, Eastern Washington University, 2015. URL <https://dc.ewu.edu/cgi/viewcontent.cgi?article=1088&context=theses>. Acessado em: 24 maio 2025.
- Kane Partridge, Jason Hayes, Daniel James, and Allan Hahn. A wireless-sensor scoring and training system for combative sports. In Proceedings of SPIE–the International Society for Optical Engineering, volume 5649, pages 402–408. SIPE, 2005.
- Marek Babiuch, Petr Foltýnek, and Pavel Smutný. Using the esp32 microcontroller for data processing. In 2019 20th International Carpathian Control Conference (CarpathianCC), 2019. URL <https://doi.org/10.1109/CarpathianCC.2019.8765944>.
- Ege Ciklabakkal, Ataberk Donmez, Mert Erdemir, Emre Suren, Mert Kaan Yilmaz, and Pelin Angin. Artemis: An intrusion detection system for mqtt attacks in internet of things. In 2019 38th Symposium on Reliable Distributed Systems (SRDS), 2019. URL <https://doi.org/10.1109/SRDS47363.2019.00053>.
- Espressif Systems. Esp-now for outdoor applications. <https://developer.espressif.com/blog/esp-now-for-outdoor-applications/>, 2024.
- Electric UI. Wireless protocol latency comparison. <https://electricui.com/blog/latency-comparison>, 2020.

- ESP32 Community. Esp32 power instability and current spikes in battery operation. <https://esp32.com/viewtopic.php?t=41673>, 2024.
- MySensors Forum. Looks as though esp-now is finally working, 2020. URL <https://forum.mysensors.org/topic/11727/looks-as-though-esp-now-is-finally-working/6>.
- Rui Santos and Sara Santos. Esp32 battery power: Tp4056 charger, protection circuits, and 18650 cells. <https://randomnerdtutorials.com/esp32-battery-charging-tp4056/>, 2023. Exemplos práticos de uso. Acessado em: 7 dez. 2025.
- Espressif Systems. Esp32 hardware design guidelines: Power supply and ldo recommendations. <https://docs.espressif.com/projects/esp-hardware-design-guidelines/en/latest/esp32/>, 2022. Recomendações oficiais de alimentação do ESP32. Acessado em: 7 dez. 2025.
- Worldsemi. Ws2812b intelligent control rgb led datasheet. <https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf>, 2020. Datasheet oficial do LED utilizado no módulo WCMCU-2812B-16. Acessado em: 7 dez. 2025.
- Adafruit Industries. Adafruit neopixel RGB led library. https://github.com/adafruit/Adafruit_NeoPixel, 2024. Acessado em: 7 dez. 2025.
- MicroPython Project. Neopixel library for micropython (ws2812 / rgb leds). <https://docs.micropython.org/en/latest/library/neopixel.html>, 2024a. Acessado em: 7 dez. 2025.
- Maxim Integrated. Max7219/max7221: Serially interfaced, 8-digit, led display drivers. <https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf>, 2019. Datasheet oficial do controlador MAX7219. Acessado em: 7 dez. 2025.
- Adafruit Learning System. Max7219 led matrix module guide. <https://learn.adafruit.com/adafruit-led-backpack/>, 2024. Tutorial de uso prático com microcontroladores. Acessado em: 7 dez. 2025.
- MicroPython Project. Softspi class — software-based spi interface. <https://docs.micropython.org/en/latest/library/machine.SoftSPI.html>, 2024b. Documentação oficial do MicroPython. Acessado em: 7 dez. 2025.
- mcauser. micropython-max7219: Micropython driver for max7219 8x8 led matrices. <https://github.com/mcauser/micropython-max7219>, 2017. Acessado em: 6 dez. 2025.
- NanJing Top Power ASIC Corp. Tp4056: 1a standalone linear li-ion battery charger with thermal regulation. https://www.lcsc.com/datasheet/lcsc_datasheet_

[1810271810_Nanjing-Top-Power-ASIC-Corp-TP4056_C16581.pdf](#), 2019. Datasheet oficial. Acessado em: 7 dez. 2025.

Texas Instruments. Tlv755p: 500-ma, low-dropout voltage regulator. <https://www.ti.com/lit/ds/symlink/tlv755p.pdf>, 2022. Datasheet oficial. Acessado em: 7 dez. 2025.

International Fencing Federation. Material rules. <https://static.fie.org/uploads/26/131720-book>

International Fencing Federation. Technical rules. <https://static.fie.org/uploads/26/131735-technical>

Microchip Technology Inc. Understanding brown-out conditions in microcontrollers. <http://ww1.microchip.com/downloads/en/AppNotes/01146A.pdf>, 2018. Aplicação técnica descrevendo brownout, causas e efeitos. Acessado em: 7 dez. 2025.