

Daniel Cogo de Vargas

***Software para Monitoramento de Atividades de
Natação com RFID ativo por meio das placas nRF51
da Nordic***

São José – Santa Catarina

Agosto / 2014

Daniel Cogo de Vargas

***Software para Monitoramento de Atividades de
Natação com RFID ativo por meio das placas nRF51
da Nordic***

Monografia apresentada à Coordenação do
Curso Superior Tecnológico em Sistemas de
Telecomunicações do Instituto Federal de Santa
Catarina para a obtenção do diploma de
Tecnólogo em Sistemas de Telecomunicações.

Orientador:

Prof. Ms. Marcio Henrique Doniak

Co-orientador:

Prof. Dr. Alexandro Andrade

CURSO SUPERIOR TECNOLÓGICO EM SISTEMAS DE TELECOMUNICAÇÕES
INSTITUTO FEDERAL DE SANTA CATARINA

São José – Santa Catarina

Agosto / 2014

Monografia com o título de *Software para Monitoramento de Atividades de Natação com RFID ativo por meio das placas nRF51 da Nordic*, apresentada por Daniel Cogo de Vargas em 26 de agosto 2014 , em São José, Santa Catarina, pela banca assim constituída:

Prof. Ms. Marcio Henrique Doniak
IFSC

Prof. Dr. Alexandro Andrade
UDESC/IFSC

Prof. Dr. Eraldo Silveira e Silva
IFSC

Prof. Ms. Arliones Stevert Hoeller Junior
IFSC

Resumo

As atividades de natação ainda carecem de um produto que monitore e registre o treinamento dos nadadores remotamente com a menor interferência possível. Já existe no mercado alguns produtos com esta proposta, mas nenhuma delas é uma solução dominante amplamente aceita pelos nadadores, já que a grande maioria não as utilizam. Este projeto vem propor uma nova forma de monitorar treinos de natação, acompanhando a quantidade de voltas e o tempo de cada volta percorrida pelo nadador através da Identificação por Radiofrequência (RFID) na frequência de 2,4 GHz. O RFID nesta faixa de frequência será implementado no *kit* de desenvolvimento da *Nordic* onde o leitor será feito na placa PCA10000, utilizando as placas do SoC (*system-on-chip*) nRF51, para aplicações de potências muito baixas. Neste sistema, chamado de Monitoramento de Atividades de Natação (MAN), o nadador nada com uma etiqueta de RFID que transmite seu número de identificação (ID) único para o leitor, e é através deste ID que será contabilizada as voltas e registrado o tempo de cada uma. O MAN passou por apenas um teste de integração em campo, mas os resultados obtidos já revelam que esta é uma forma promissora dos atletas de natação terem um resultado completo dos seus treinos de natação, podendo avaliar parâmetros como: distância percorrida, tempo de cada volta, tempo total, velocidade média de cada volta e do treino, entre outros.

Palavras-Chave

Natação, RFID, nRF51, *Nordic*

Abstract

The swimming activities still require a product that monitor and record the training of swimmers remotely with the least possible interference. There is already some products on the market with this proposal, but neither is a dominant solution widely accepted by swimmers, since the vast majority do not use it. This project proposes a new way to monitor swimming training, tracking the number of turns and the time for each lap walked by the swimmer through the Radio Frequency Identification (RFID) in the 2.4GHz frequency. RFID in this frequency range will be implemented in the Nordic development kit where the reader will be done on the board PCA10000 using boards SoC (system-on-chip) nRF51, for very low power applications. In this system, called Activity Monitoring Swimming (MAN), the swimmer swims with an RFID tag that transmits to the reader his unique identification number (ID), and it is through this that ID will be counted laps and recorded the time each. The MAN just went through an integration test on, but the results already reveal that this is a promising way of swimmers have a complete result of their swimming training and can evaluate parameters such as distance traveled, time of each lap , total length, average speed and lap on each training, among others.

Keywords

Swimming, RFID, nRF51, *Nordic*

Sumário

Lista de Figuras

Lista de Tabelas

1	Introdução	p. 10
1.1	Atividades de Natação	p. 10
1.2	Proposta	p. 12
1.3	Metodologia	p. 13
1.4	Organização do texto	p. 15
2	Fundamentação Teórica	p. 16
2.1	Treinamento de Natação	p. 16
2.2	Equipamentos para Acompanhamento de Natação	p. 17
2.2.1	Forerunner 910XT	p. 17
2.2.2	Sensores de Toque	p. 19
2.3	RFID no Monitoramento de Atividades de Natação	p. 20
2.4	RFID	p. 22
2.4.1	Leitor de RFID	p. 22
2.4.2	Tag	p. 22
2.4.3	Middleware	p. 24
2.5	Kit de desenvolvimento NRF51	p. 25
2.6	Daemon	p. 25
2.6.1	Python	p. 26

2.7	Apache	p. 26
2.7.1	PHP	p. 27
2.7.2	JQuery	p. 28
2.8	Banco de Dados	p. 28
2.8.1	MySQL	p. 28
2.8.2	MER - Modelo Entidade Relacionamento	p. 29
3	MAN - Monitoramento de Atividades de Natação	p. 31
3.1	Daemon no MAN	p. 32
3.2	O Banco de Dados do MAN	p. 33
3.2.1	Tabela de Piscina	p. 34
3.2.2	Tabela de Usuário	p. 35
3.2.3	Tabela de Treino Proposto	p. 35
3.2.4	Tabela de Treino Realizado e Tabela de Voltas	p. 36
3.2.5	Diagrama de Entidade Relação do MAN	p. 38
3.3	Interface com o Usuário	p. 39
3.3.1	Configuração do Sistema	p. 40
3.3.2	Cadastro de Usuário	p. 41
3.3.3	Cadastro de Treino	p. 41
3.3.4	Cadastro de Piscina	p. 41
3.3.5	Realização do Treino	p. 42
4	Testes	p. 47
4.1	Informações Básicas Para Realização dos Testes	p. 47
4.2	Teste de Laboratório	p. 48
4.3	Testes de Campo	p. 48
4.4	Teste de Validação do Daemon	p. 51

4.5	Teste de Acompanhamento de Treino	p. 51
4.6	Limitações	p. 54
5	Conclusões	p. 56
5.1	Trabalhos Futuros	p. 57
	Lista de Abreviaturas	p. 59
	Referências Bibliográficas	p. 61

Lista de Figuras

2.1	Dispositivo <i>Forerunner 910XT</i> comercializado pela <i>Garmin</i> (GARMIN, 2013).	p. 18
2.2	Sensor de toque, que localiza-se na borda dentro da piscina (ROMERO, 2012).	p. 19
2.3	Bloco de partida com sensor, que localiza-se na borda da piscina (ROMERO, 2012).	p. 20
2.4	Chegada do nadador proximo ao leitor de RFID. (RAULINO, 2013)	p. 21
2.5	Virada do nadador proximo ao leitor de RFID. (RAULINO, 2013)	p. 21
2.6	Modo de operação do leitor de RFID (DONIAK; GREFF, 2009).	p. 23
2.7	Etiqueta (SANGHERA, 2007).	p. 23
2.8	Exemplo de um Diagrama Entidade Relação (HEUSER, 1998).	p. 30
3.1	Diagrama de blocos do sistema MAN.	p. 32
3.2	Fluxograma de operação do <i>daemon</i>	p. 33
3.3	Diagrama entidade relação do MAN, criado com auxílio do programa <i>MySQL Workbench</i>	p. 40
3.4	Diagrama de blocos do programa de acompanhamento de treinos.	p. 42
3.5	Diagrama de blocos do processo de registro na tabela <i>voltas</i> .	p. 45
4.1	Leitura da <i>tag</i> sendo exibida em um terminal.	p. 49
4.2	Foto exibindo o treino, com potência de -20dBm e a <i>tag</i> na touca.	p. 50
4.3	Resultado das inserções na tabela <i>serial</i> .	p. 52
4.4	Teste de acompanhamento do treino de natação.	p. 54

Lista de Tabelas

2.1	Tabela comparativa entre o trabalho desenvolvido por <i>Mario Felipe Raulino</i> e a proposta deste trabalho.	p. 21
3.1	Tabela com valores e descrições da tabela utilizada pelo <i>daemon</i>	p. 33
3.2	Valores e descrições da tabela <i>piscina</i>	p. 35
3.3	Valores e descrições da tabela de <i>Usuario</i>	p. 36
3.4	Valores e descrições da tabela de treino proposto.	p. 37
3.5	Valores e descrições da tabela de treino a ser realizado.	p. 38
3.6	Valores e descrições da tabela <i>voltas</i>	p. 39
4.1	Valores dos testes de alcance.	p. 51
4.2	Valores de acompanhamento de treino.	p. 53

1 Introdução

O presente trabalho traz uma proposta de monitoramento remoto de atividades de natação através da tecnologia *Radio-Frequency Identification* (RFID) utilizando o *kit* de desenvolvimento nRF51, da fabricante *Nordic Semiconductors*. RFID é a tecnologia que visa transferir informações através de ondas de rádio. As informações ficam contidas em uma etiqueta ou *tag* que são lidas por um dispositivo chamado leitor de RFID. O emprego da tecnologia RFID utilizada será de alta frequência, que visa aumentar o alcance do sinal e diminuir o tamanho das antenas, obtido com o emprego do *kit* da *Nordic*. Também é desenvolvido um *software* de acompanhamento de treinos de natação, fazendo uso de PHP e *MySQL*.

O Monitoramento de Atividades de Natação (MAN), consiste em registrar a quantidade de voltas e o tempo de cada volta (DONIAK; GREFF, 2009). Neste sistema, o nadador utiliza *tags* que são lidas pelo leitor de RFID cada vez que completa uma volta. Através desta leitura é possível saber o tempo que completou a volta e salvar em um banco de dados para futuras consultas.

Para isso é necessário entender como são planejados os treinos e desenvolver um *software* que, além de registrar a quantidade de voltas e seus respectivos tempos, calcula também as informações relevantes do treino.

1.1 Atividades de Natação

Segundo o Atlas dos Esportes (NOLESCO; PAVEL; MOURA, 2006), a natação é um esporte aquático que tem como objetivo imediato, para o atleta, vencer uma determinada distância em meio líquido ao menor tempo possível.

Para que atletas consigam um alto rendimento em competições ou que nadadores amadores possam melhorar seu desempenho na prática do esporte, são realizados vários tipos de treinamentos desportivos. Estes treinamentos desportivos são processos pedagógicos que envolvem a preparação física, técnica, tática, teórica e psicológica do atleta (FREITAS, 1999).

Para que atletas e treinadores consigam mensurar o desempenho durante os treinamentos, são realizados acompanhamentos dos treinos analisando o tempo que o nadador leva para completar uma volta. Com o acompanhamento dos treinos é possível analisar a evolução do atleta, ajudando-o a mantê-lo motivado, além de, controlar e evitar, por exemplo, o *overtraining*¹.

A maneira mais comum de monitorar o acompanhamento dos treinos é através de cronômetro, onde o tempo é medido de maneira manual, pelo treinador ou pelo próprio atleta, a cada volta ou conjunto de voltas que o atleta dá na piscina.

Atualmente já é possível comprar relógios (GARMIN, 2013) que são capazes de monitorar os treinos de natação. Estes funcionam como os pedômetros na corrida, pois assim como na corrida precisa calibrar o tamanho da sua passada, na natação precisa calibrar a braçada de cada estilo de nado. Porém, este tipo de relógio apresenta alguns problemas, por exemplo, não ser capaz de registrar as voltas quando o nadador está treinando apenas as batidas das pernas. E também, o nadador deve seguir o padrão de braçadas para que não haja variação na medida. Só que o padrão de braçada pode mudar conforme o treino, por exemplo, em um treino de explosão as braçadas tem uma amplitude diferente de um treino de resistência, ocasionando erros significativos das medidas obtidas.

Outra forma de medir o tempo de cada volta é através de sensores de toque (ROMERO, 2012) que é o método de aferição utilizado em competições organizados pela Federação Internacional de Natação (FINA) e Confederação Brasileira de Desportos Aquáticos (CBDA). Porém, este método é muito caro, sensível e se limita a pelo menos um nadador por raia.

Poucos clubes e academias oferecem uma estrutura adequada para monitoramento remoto de treinos. Geralmente a contagem de voltas é feita pelo próprio nadador, fazendo com que ele diminua a concentração no seu nado para memorizar as voltas dadas em seus treinos. Ao longo do treinamento o cansaço torna mais difícil à contagem de voltas. Logo, o atleta ficará com dúvida sobre o quanto de fato nadou. Outro problema para os atletas é a contagem do tempo, no caso do nadador precisar, além de contabilizar as voltas dadas na piscina, registrar o tempo médio de cada volta, a dificuldade aumenta e conseqüentemente, a sua concentração no nado diminuirá ainda mais. Estas formas de medida da quantidade de voltas, conjunto de voltas e dos seus respectivos tempos, torna o planejamento dos próximos tempo sempre subjetivos, baseados apenas nas distância que será percorrida e em uma intensidade subjetiva julgada pelo próprio nadador.

Normalmente, junto as piscinas tem um cronômetro grande analógico, que auxilia os na-

¹Overtraining – é o desequilíbrio entre estresse e recuperação que acarreta a queda no rendimento do atleta, o qual pode se prolongar por meses

dadores a cronometrarem seu tempo. Porém, o nadador não pode parar a cada virada para olhar o relógio. Assim, ele acaba verificando apenas o tempo de uma série de voltas e perde a medida de cada volta. Este processo é uma forma subjetiva do nadador ter uma ideia de seu desempenho. Uma forma de fazer a medida de cada volta dada é pedir para alguém ficar observando e anotando o tempo de cada volta. Mas nem mesmo, as equipes de natação com uma boa infraestrutura conseguem proporcionar tal forma de acompanhamento de treino, porque não há atleta disponível para cada atleta que está nadando.

Devido as circunstâncias citadas, um sistema automatizado que realize o acompanhamento remoto do treino, poderia auxiliar nadadores profissionais e amadores, a manterem-se sempre informados sobre o resultado de cada treino. Podendo, também, consultar a evolução em seus treinos de natação, para que busquem uma melhoria contínua.

O Monitoramento de Atividades de Natação (MAN) é um sistema que faz o acompanhamento dos treinos de natação utilizando o sistema RFID. Onde o nadador utiliza sensores (chamadas de *tags* ou etiquetas) e na borda da piscina há um equipamento que faz a leitura da passagem do nadador pela borda, marcando assim, o tempo de cada passagem. Desta maneira pode-se obter o tempo que leva de uma volta a outra. Outros trabalhos já descreveram o desenvolvimento deste sistema (RAULINO, 2013).

O trabalho de (RAULINO, 2013) descreve o desenvolvimento de um sistema de acompanhamento de treinos de natação através do sistema RFID em baixa frequência (125KHz), obtendo um alcance curto entre o sensor utilizando pelo nadador e o leitor que se localiza na borda da piscina, fazendo com que algumas passagens do nadador pelo leitor não sejam lidas.

Além deste problema, (RAULINO, 2013) desenvolveu um software de acompanhamento em *Java* embarcado em uma placa *Raspberry Pi*. Pelo fato de utilizar *Java Virtual Machine* - Máquina Virtual Java (JVM) e da placa ter uma *Random Access Memory* - Memória de Acesso Aleatório (RAM) muito limitada (256 KB), relatou (RAULINO, 2013), um alto consumo de RAM da placa durante a utilização do *software* de acompanhamento do treino de natação.

1.2 Proposta

A proposta de estudo deste trabalho é o desenvolvimento de uma nova forma para acompanhamento de treinos de natação. Este sistema foi desenvolvido utilizando PHP, e além de acompanhar o treino do nadador, irá salvar os dados coletados em um banco de dados, para que o nadador possa acompanhar o treinos realizados e manter-se sempre atualizado.

Foi escolhido utilizar PHP para o desenvolvimento devido ao fato de ser linguagem simples para desenvolver, com bastante potencial e pelo fato de ser executado dentro de um servidor HTTP, sendo assim, o programa pode ser acessado de qualquer computador dentro de uma rede de computadores ou até mesmo fora desta rede, assim o usuário ou o treinador podem acessar os dados mesmo não estando no local onde foi instalado o MAN. O PHP também possui facilidade de interação com o *MySQL*, que é o Sistema de Gerenciamento de Banco de Dados (SGBD) utilizado no sistema. E ao contrário da máquina virtual *Java*, usada por (RAULINO, 2013), o servidor que roda o PHP não apresenta um crescimento contínuo de sua memória RAM durante o funcionamento do programa MAN. Além disso, pode operar em diversos sistemas operacionais e diversos tipos de máquinas, até em sistemas embarcados, como por exemplo, em um *Raspberry Pi*.

Para forma de realizar a marcação do tempo de cada volta é feita através da tecnologia RFID, sendo utilizado o *kit* de desenvolvimento nRF51, da fabricante *Nordic*, que opera com a frequência de 2.4 GHz, podendo conseguir um longo alcance entre o sensor (*tag*) e o leitor localizado na borda da piscina.

1.3 Metodologia

Antes de descrever a maneira de como desenvolver o MAN é necessário entender seu funcionamento, o cenário onde será implementado e como realiza as medidas.

O MAN é um sistema, para acompanhar treinamentos de natação, que faz utilização da tecnologia RFID, com *tags* ativas para aferir o tempo que um nadador pode completar uma volta. O nadador utiliza duas *tags* de RFID ativas, uma no punho e uma no tornozelo. Toda vez que este passa na borda da piscina, onde localiza-se o leitor de RFID é feita a leitura do código de identificação da *tag* e o *time stamp* do momento da leitura. Este valor de *time stamp* é utilizado para realizar a contagem do tempo da volta, o sistema marca o *time stamp* da abertura e chegada da volta, subtraindo uma pela outra, descobrindo assim o tempo, em segundos, que levou para completar a volta.

A finalidade da *tag* do tornozelo é abrir uma série e a finalidade da *tag* do punho é para registrar o momento da chegada, pois o braço se próximo do leitor no momento do leitor no momento da chegada da volta, este modelo de sistema utilizando duas *tags* foi implementado em trabalhos anteriores *mario*. Estas *tags* ativas, transmitem o código de identificação ao leitor de RFID em uma sequência de vários pacotes, porem o sistema necessita que ocorra somente a leitura de um pacote, para que o sistema não faça a leitura de mais um pacote e contabilize

a volta de maneira errônea, foi estipulado um tempo de dez segundos após a leitura do pacote, para que o nadador consiga sair da área de cobertura do leitor de RFID.

Como leitor de RFID e *tag*, foi utilizado o *kit* de desenvolvimento nRF51, da fabricante *Nordic*, este *kit* possui grande capacidade, pois pode operar tanto como transmissor (*tags* ativas) como receptor (leitor de RFID), opera na frequência de 2.4GHz, ideal para utilização do MAN, pois uma das propostas é desenvolver o sistema operando com RFID de alta frequência.

O MAN também possui um sistema de acompanhamento de treinos, onde um programa desenvolvido em *python* faz a leitura da serial onde está conectado o leitor de RFID. Este programa passa as informações da leitura da *tag* para um outro programa em desenvolvimento em PHP, que faz o acompanhamento do treino, contabilizando as voltas, séries e intervalo de descanso. Para realizar a passagem dos dados lidos pelo leitor, pelo programa em *python* para o programa em PHP é utilizada uma tabela de banco de dados, o programa em *python* insere os valores da leitura nesta tabela e o programa em PHP consulta as inserções nesta tabela para poder realizar o procedimento de acompanhamento de treino.

Para o sistema poder registrar as voltas dos treinos, deve possuir uma tabela de banco de dados onde ficam salvos as voltas realizadas, porém estas voltas devem ficar associadas ao nadador que está realizando o treino, por isso é necessário desenvolver uma tabela para identificar de qual usuário é o treino em questão e uma outra tabela para saber de qual nadador pertence a volta. Também é importante criar uma tabela onde estará os treinos propostos pelo treinador ou usuário. Sendo esta situação deve existir a tabela onde fica cadastrado o usuário, a tabela onde ficam os treinos propostos pelo usuário e a tabela onde estão os treinos feitos pelo usuário e uma tabela para identificar de qual usuário pertence a volta.

Para utilizar o sistema, primeiramente o usuário deve realizar cadastro no sistema, com alguns dados necessários para operação do sistema, como por exemplo as *tags* que utilizará. Em seguida deve realizar o cadastro dos treinos que irá realizar. No momento da realização do treinamento, deve selecionar o treino que deseja realizar dentre os que foram escolhidos para o mesmo. Todo este procedimento é feito através do programa em PHP que pode ser acessado através de um navegador, sendo executado em um monitor próximo a piscina.

Após selecionar o treino que irá realizar, deve pressionar o botão de início para começar a realizar a leitura das voltas. Por se tratar de um protótipo, durante os testes terá um computador próximo a piscina, e a execução de início do treino será pressionado pela pessoa que estiver operando o computador durante o treino.

Por fim, após desenvolver o sistema e realizar testes em laboratório, foram realizados testes

para validar o funcionamento do MAN em uma piscina. Foi escolhida a piscina do CEFID, que é um campus da UDESC, que possui uma dimensão semiolímpica, com comprimento de 25 metros. Foram realizado testes visando a distância entre o leitor e a *tag*, alterando a potência para chegar a um alcance aceitável. Também foi realizado um teste para medir o tempo do treino e comparar com os valores medidos em um cronometro.

1.4 Organização do texto

O texto a seguir está organizado em fundamentação teórica, desenvolvimento, testes e conclusão. No Capítulo 2 é apresentada a fundamentação teórica, onde são descritas as ferramentas utilizadas para o desenvolvimento do MAN, como o RFID e o *kit* de desenvolvimento da *Nordic*, também são descritas as linguagens utilizadas (como o *Python* e o *PHP*) e, além disso, são apresentados o banco de dados *MySQL* e o Modelo Entidade Relacionamento (MER). No Capítulo 3 é abordada a maneira de como foi desenvolvido o sistema MAN, descrevendo as tabelas do banco de dados e a lógica do acompanhamento do treinos. No Capítulo 4 é apresentado os testes realizados para averiguar o funcionamento do sistema. No Capítulo 5 são apresentadas as conclusões e sugestões para trabalhos futuros.

2 *Fundamentação Teórica*

Neste capítulo são descritas as referências teóricas para o desenvolvimento do MAN. Inicialmente são abordados alguns sistemas comerciais que possuem funcionalidade semelhante ao sistema proposto. Em seguida, são descritas as tecnologias necessárias para o desenvolvimento do MAN.

O MAN, consiste de um *software* e um *hardware* que faz o acompanhamento de um treino de natação, desenvolvido em PHP, registrando o tempo e o número da volta. A partir daí é possível calcular outros valores do treino como a velocidade. Também é possível salvar estes dados em um banco de dados, *MySQL*. Este programa considera a quantidade de séries, voltas por séries e o tempo de descanso entre os intervalos de cada série. Um sistema RFID marca o tempo da volta completada pelo nadador, para o sistema de RFID foi utilizado o *kit* de desenvolvimento da *Nordic*.

O dispositivo da *Nordic* (leitor de RFID) é conectado ao USB de um computador, onde um programa desenvolvido em *Python* recebe o código da *tag* lida e repassa para uma tabela de banco de dados. Um programa desenvolvido em *jQuery* e em PHP recebe os registros feitos nesta tabela e faz o tratamento das informações, séries, volta por série e tempo de descanso.

Este capítulo faz a abordagem de cada uma das ferramentas utilizadas.

2.1 **Treinamento de Natação**

Segundo Rödel (RODEL, 2011), para um treino de natação ser planejado devem-se levar em consideração alguns fatores como a duração (tempo de treino por dia), frequência (dias de treino por semana), intensidade (velocidade dos treinos) e volume (distância percorrida durante os treinos).

Estes fatores podem incidir no tipo de treino planejado para o nadador. Os tipos de treinos podem ser:

- Treino intervalado: são repetições em séries intervaladas, com esforço e repouso controlados.
- Treinamento de repetições: treino intervalado, composto de séries curtas e com velocidades maiores do que as utilizadas nas provas, intercalado com grandes intervalos de descanso.
- Treinamento de velocidade crescente: várias séries com repouso pequenos, não permitindo um total recuperação.
- Treinamento de velocidade decrescente: Treinamento onde o nadador começa com uma velocidade máxima e vai diminuindo esta velocidade gradativamente até o final do treino.

2.2 Equipamentos para Acompanhamento de Natação

Não existem produtos dominantes no mercado que possam auxiliar treinadores, professores, atletas e praticantes da modalidade que possam acompanhar e salvar os dados do treino. Entre os mais conhecidos está o *Forerunner 910XT* desenvolvido pela *Garmin*, este produto é muito conhecido por triatletas, pois pode, monitorar natação, corrida e ciclismo. Existem também os sensores de toque, que são utilizados em competições para cronometrar o tempo entre o salto na piscina e o fim da prova.

2.2.1 Forerunner 910XT

É um dispositivo para auxílio de treinamento para esportes na terra e na água, muito utilizado por triatletas. Para esportes na água consegue fazer o registro da distância nadada, o tempo, a velocidade e é capaz de estimar a eficiência do treino. Como dados de entrada deve definir o tipo de nado e a dimensão da piscina (GARMIN, 2013).

Para a prática de natação possui dois modos de operação, natação em águas abertas (ou natação em longa distância) e natação em piscina. Este aparelho possui alguns botões que devem ser pressionados durante o exercício em piscina, para começar o exercício deve pressionar o botão *start*, para os intervalos (descanso) de natação deve ser pressionado o botão *lap*, tanto para entrar e sair do intervalo, para terminar o treino deve ser selecionado o botão *stop*. O *display* do dispositivo apresenta três informações: tempo, distância e ritmo. O *Forerunner 910xt* pode ser visto na Figura 2.1.

Este aparelho considera cinco estilos de nado (tipo de braçada) que são *crawl*, costas, peito,



Figura 2.1: Dispositivo *Forerunner 910XT* comercializado pela *Garmin* (GARMIN, 2013).

borboleta e misto. Este último é definido como sendo mais de um estilo de nataação em um intervalo de tempo.

Antes de iniciar o treino, algumas informações devem ser definidas no dispositivo. O *Forerunner* possui um *menu* de navegação, onde antes de iniciar o treino, deve ser escolhida a modalidade esportiva que vai realizar (ciclismo, corrida ou nataação), se escolher a nataação, em seguida, o atleta deverá selecionar um dos cinco tipos de estilo de nado disponíveis. Antes de realizar o treino, o atleta deve estar ciente que o tamanho da piscina onde o treino será realizado já está configurado.

Este dispositivo possui uma função interessante que pode ser útil para muitos nadadores, que é a sua capacidade em contar a quantidade de braçadas dadas a cada volta. Considerando uma braçada sempre que o braço em que esta sendo usado completa um ciclo completo.

Garmin Connect

Garmin Connect é um *web site* ou uma comunidade virtual onde é possível fazer o levantamento dos dados estatísticos do treino. As informações que chegam ao computador, pelo *Forerunner 910XT*, são enviadas ao *Garmin Connect* através do programa *ANT Agent*¹.

Nesta comunidade é possível visualizar o tempo de treino, os tempos de cada intervalo, e um gráfico de velocidade por tempo. Além disso, pode compartilhar informações com outros membros desta comunidade. É no *Garmin Connect* que as informações ficam armazenadas

¹<http://connect.garmin.com>. Acessado em 7 de julho de 2013

e podem ser consultadas pelo usuário a qualquer momento, mesmo que sejam apagadas do *Forerunner 910xt*.

No *Garmin Connect* também é possível analisar o desempenho do atleta, tanto no treino realizado como ao longo de um período maior. Ele permite que o atleta visualize seus resultados através de gráficos que facilita o entendimento.

O lado negativo é o fato de calibrar as braçadas que implicam em um erro imensurável. Outro ponto negativo, é o nadador ter que selecionar o estilo do nado cada vez que alterá-lo durante o treino. Também existe o fato do nadador treinar apenas pernas, neste caso, não será contabilizado as voltas dadas na piscina.

O principal ponto positivo deste dispositivo é que ele é de uso pessoal. Ele não precisa ser oferecido pela academia . Assim, o atleta não depende de uma infraestrutura para acompanhar o seu treino, isso é uma escolha que ele mesmo pode fazer.

2.2.2 Sensores de Toque

Como foi informado estes sensores são eletro-mecânicos e são acionados de maneira manual. Estes sensores são chapas que podem ser fixadas no bloco de partida conforme a Figura 2.2. Ou também, podem ser construídas no próprio bloco conforme a Figura 2.3. Estes sensores são comuns em competições organizadas pela CBDA e pela FINA.

O nadador fica pressionando o sensor na plataforma de salto, quando é dada a largada ele salta na piscina e inicia a contagem de tempo. A pressão no sensor serve para verificar se algum nadador saltou antes de ter sido dada a largada. Pois, assim que o nadador salta para a piscina a pressão no sensor cessa e é registrado o momento da partida. Em cada da borda da piscina é fixado um sensor por raia, que oficialmente é de 2.5m. Estes sensores são responsáveis por registrar o tempo do nadador cada vez que ele passa por uma das bordas da piscina.

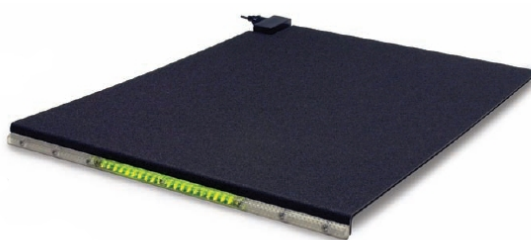


Figura 2.2: Sensor de toque, que localiza-se na borda dentro da piscina (ROMERO, 2012).

O método de aferição destes sensores pode variar, dentre os diversos fabricantes, por exem-



Figura 2.3: Bloco de partida com sensor, que localiza-se na borda da piscina (ROMERO, 2012).

plô, os sensores fabricados pela *Colorado Timing System* é por peso e pressão, enquanto os da *Omega Timing* utiliza força de impulsão e pressão (ROMERO, 2012).

Além do registro de tempo e contagem de voltas, este dispositivo é capaz de determinar vencedores em uma competição. Por exemplo, nas Olimpíadas de Pequim, *Michael Phelps* ganhou uma de suas oito medalhas de ouro, com a diferença para o segundo colocado sendo definida na intensidade do toque na chegada. Os dois bateram no mesmo tempo, mas foi *Phelps* quem tocou com mais força no sensor e assim, foi determinado campeão (SALETAN, 2008).

Comparado ao MAN, o sensor permite que apenas que um nadador nade por raia. O MAN não apresenta esta restrição, já que cada nadador é identificado pelo dispositivo. Porém, no MAN o nadador deve portar pelo menos uma etiqueta (*tag*), o que não ocorre no sensor de toque.

2.3 RFID no Monitoramento de Atividades de Natação

O *RaspberryPi e RFID passivo no Monitoramento de Atividades de Natação* (RAULINO, 2013) foi um trabalho que tinha por objetivo utilizar um leitor de RFID, em uma das bordas da piscina. Neste trabalho o equipamento de RFID utilizado foi desenvolvido para controle de acesso e, apesar de ser especificado para atingir um alcance de até 1 metro, seu alcance real foi de apenas 50cm. Sua faixa de frequência é de 125KHz. Nesta faixa de frequência para se conseguir um alcance grande, deve-se aumentar o tamanho da antena e o leitor de RFID fica muito grande, pesado e caro, inviabilizando para este projeto.

Outro objetivo do trabalho de *Mário Felipe Raulino* (RAULINO, 2013) era criar um sistema embarcado. Para isso, ele usou uma placa *Raspberry Pi* com 256MB de memória RAM. O

sistema de monitoramento foi desenvolvido em *Java*, havendo um grande consumo de mem ria RAM do *Raspberry Pi*, devido a utiliza o da JVM (RAULINO, 2013).

Comparado ao trabalho de *M rio Felipe Raulino* (RAULINO, 2013), para o MAN ser  utilizado um sistema de RFID ativo que opere na faixa de 2.4GHz, assim ser  poss vel ter um alcance maior entre a *tag* e o leitor. Para o programa de acompanhamento do do treino de nata o, o MAN   desenvolvido em PHP, para tentar evitar a utiliza o elevada da RAM.

Enquanto o nadador utiliza duas *tags* de RFID, uma no pulso e outra no tornozelo, sendo que cada *tag* possui um n mero de identifica o  nico. A *tag* do pulso serve, principalmente, para abrir e finalizar uma sequ ncia de voltas, conforme a Figura 2.4, enquanto a *tag* do tornozelo, normalmente,   usada para finalizar uma volta e imediatamente abrir uma nova volta logo ap s a virada, conforme a Figura 2.5.



Figura 2.4: Chegada do nadador proximo ao leitor de RFID. (RAULINO, 2013)



Figura 2.5: Virada do nadador proximo ao leitor de RFID. (RAULINO, 2013)

A Tabela 2.1 apresenta as diferen as entre o trabalho desenvolvido por *Mario Felipe Raulino* (RAULINO, 2013) e a proposta deste trabalho.

Tabela 2.1: Tabela comparativa entre o trabalho desenvolvido por *Mario Felipe Raulino* e a proposta deste trabalho.

Funcionalidade	Trabalho de <i>Mario Raulino</i>	Proposta
<i>tag</i>	Passiva	Ativa
Frequ�ncia	Baixa (125KHz)	Alta (2.4GHz)
Linguagem	<i>Java</i>	PHP
Embarcado em <i>Raspberry Pi</i>	Sim	N�o
Possui banco de dados	N�o	Sim

2.4 RFID

A identificação por radiofrequência consiste transportar informações através de ondas de rádio. Este sistema é composto por três componentes: *tag* (etiqueta), o leitor de RFID e o *middleware*.

A *tag* é onde está contida as informações e por intermédio de um *hardware* (leitor de RFID) que faz a leitura desta etiqueta (FINKENZELLER, 2010). Outro componente do sistema é o *middleware*, que é um *software* responsável para passar a informação que chega pelo leitor para um outro processo que realizará o processamento desta informação.

2.4.1 Leitor de RFID

O Leitor de RFID é um *hardware* que faz a leitura da identificação da *tags* de RFID, pode operar com frequências chamadas de *Low Frequency* - Baixa Frequência (LF), com a faixa de frequência de 30 a 300 KHz, *High Frequency* - Alta Frequência (HF) com faixa de frequência de 3 a 30MHz, *Ultra High Frequency* - Ultra Alta Frequência (UHF) de 300MHz a 3GHz e *Microwave* - Microondas (MW) 2 a 30GHz. Ele gera um sinal com uma tensão que varia de 3 a 15 Volts, sendo que o nível lógico baixo (0) para a maior tensão e o nível lógico alto (1) para a menor tensão (DONIAK; GREFF, 2009).

O leitor de RFID transmite o sinal para a *tag* que retorna um sinal com as informações contidas na etiqueta.

Possui dois modos de operação, o modo biestático e o monoestático. No modo biestático, possui duas antenas, uma para transmitir o sinal para a *tag* e outra para receber o sinal de retorno da *tags*. No modo monoestático, possui uma antena só que transmite e recebe o sinal para a *tag*. A Figura 2.6 representa o processo de operação dos dois modos (DONIAK; GREFF, 2009).

2.4.2 Tag

A *tag* ou etiqueta, é um dispositivo que possui uma antena que pode ser submetida a um campo eletromagnético fazendo com que sua identificação seja transmitida para o leitor sem que haja a necessidade de fios, linha de visada ou ação de alguma pessoa. Este processo pode durar até 100 ms (ROSA, 2006).

A *tag* é dividida em 3 partes: *chip*, antena e substrato. O *chip* é um componente eletrônico feito de silício e também é dividido em 3 partes, que são a unidade lógica (responsável por pro-

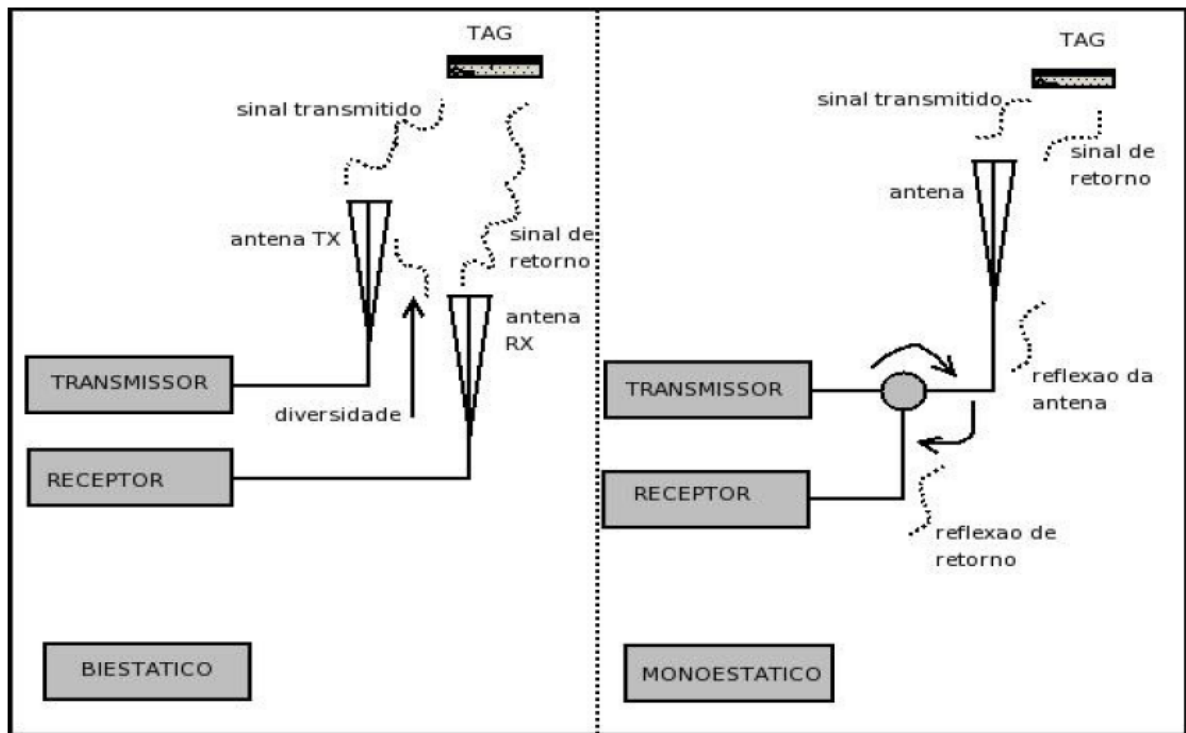


Figura 2.6: Modo de operação do leitor de RFID (DONIAK; GREFF, 2009).

cessar o protocolo entre a comunicação com o leitor), a memória (para armazenar os dados) e o modulador (para modular e/ou demodular o sinal que chega através da antena). A antena feita de metal ou material metálico, feita para receber o sinal. O substrato é a camada que abriga a antena e o *chip*. É possível visualizar a etiqueta e seus componentes na Figura 2.7 (SANGHERA, 2007).

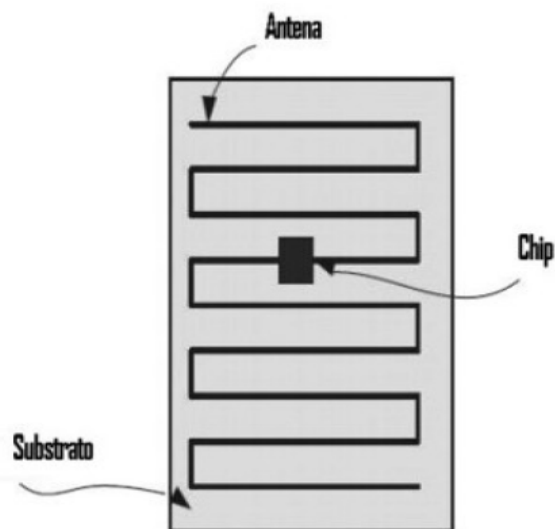


Figura 2.7: Etiqueta (SANGHERA, 2007).

As etiquetas (*tags*) podem ser do tipo passivas, ativas, semi-ativas e do tipo *Surface Acoustic*

Wave (SAW).

Tags Passivas

As *tags* passivas não possuem fonte de alimentação, sendo o seu circuito alimentado pela energia eletromagnética do leitor de RFID. As vantagens de utilizar estas etiquetas em projetos é o tamanho reduzido e o baixo custo, porém, é necessário um curto alcance entre o leitor e a *tag* (JONES; CHUNG, 2008).

Tags Ativas

As *tags* ativas possuem fonte de alimentação, a necessidade de possuir esta bateria torna a *tag* maior e com um custo maior, do que a passiva, porém, pode ter um alcance entre a *tag* e o leitor maior (JONES; CHUNG, 2008).

Tags Semi-Ativas

Esta *tag* foi desenvolvida com as características das duas *tags* citadas anteriormente, para fins de excluir as desvantagens contidas em ambas. A bateria é utilizada para alimentação do circuito, porém a comunicação com o leitor é semelhante ao modo passivo. Essa técnica prolonga a vida útil da bateria (JONES; CHUNG, 2008).

Tags SAW

Estas *tags* são semelhantes a passiva, porém, o funcionamento é diferente. Estas etiquetas convertem o sinal do leitor em ondas acústicas que passam por refletores que codificam em trem de pulsos, posteriormente convertidos em ondas de rádio para o retorno ao leitor (JONES; CHUNG, 2008).

2.4.3 Middleware

O *Middleware* é um *software* que faz a integração do fluxo de dados entre a etiqueta e um *software*, que fará o tratamento da informação. Faz a gerência dos dados coletados e também permite a integração com outros sistemas. De maneira geral o *middleware* recebe a informação coletada pelo leitor de RFID e os manipula a fim de serem utilizados um *software* (DONIAK; GREFF, 2009).

2.5 Kit de desenvolvimento NRF51

O *kit* de desenvolvimento NRF51, da fabricante *Nordic* é composto por quatro placas, são os modelos PCA10000, PCA10004, PCA10005 e *J-Link Lite Cortex M-9*. As placas PCA10000, PCA10004 e PCA10005 operam como transceptores, ou seja, podem ser tanto transmissores como receptores, portanto, podem operar tanto como leitores de RFID assim como *tags* ativas (PINHO, 2014).

A placa PCA10000 é um *dongle*, ou seja, um dispositivo que pode ficar conectado a um computador, faz isso através de uma comunicação USB que também serve de alimentação para a placa. No sistema MAN esta placa irá operar como leitor de RFID.

As placas PCA10004 e PCA10005 são placas que possuem conectores e contatos que podem ser utilizações para realizar medições dos componentes da placa, e também podem ser alimentadas pilhas de 3.3 volts. No sistema MAN esta placa irá operar como *tag* ativa.

As placas PCA10000, PCA10004 e PCA10005 possuem um *chipset* que é o NRF51822. Este componente é um sistema do tipo *System-on-chip* (SoC), que possui uma CPU de 32 bits, memória *flash* de 256 KB, 16 bits de memória do tipo RAM, transceptor de 2.4 GHz, taxa de transmissão de 2 Mbps e potência de transmissão de até 4 dBm, podendo variar entre 4 e -20dBm.

A placa *J-Link Lite Cortex M-9* serve para conectar as placas PCA10004 e 10005 ao computador, esta não possui o *chipset* NRF51822. Em um computador um programa desenvolvido em C faz a programação e depuração das placas. Com este programa é possível definir o código do ID da *tag* e a potência de transmissão do sinal.

Este *kit* foi escolhido devido a sua versatilidade, podendo operar como transceptores, e por operar na faixa de frequência de 2.4 GHz.

No sistema MAN, o leitor de RFID irá passar as informações do código do ID da *tag* para o computador, através da porta serial. Um *daemon*, fica aguardando a chegada dos dados vindos pelo leitor, para, em outro nível do programa poder realizar o processo de acompanhar o treino, registrando a volta.

2.6 Daemon

Os *Daemons* são programas que utilizam tarefas de monitoramento de um servidor ou computador. Estes programas operam em *background*, ou seja, operam de maneira despercebida

pelo usuário e não devem possuir terminal associado. Geralmente são inicializados pelo sistema operacional durante o *boot* e executados constantemente a espera de algum evento significativo para que possam executar uma ação. Estes programas devem ser executados enquanto o sistema estiver ativo e encerrado junto com os mesmos (LAUERANO, 2005).

2.6.1 Python

Python é uma linguagem de programação criada em 1990 por *Guido Van Rossum*, sendo originalmente uma linguagem voltada para usuários como físicos e engenheiros. Esta linguagem teve origem de outra, chamada ABC (BORGES, 2010).

Algumas de suas principais características são: o fato de não necessitar pré-declarar variáveis, sendo determinadas dinamicamente, o controler de blocos é feito por indentação, ou seja, não há necessidade de caracteres delimitadores (como {}, por exemplo) ².

Python é um tipo de linguagem interpretada, ou seja, ela é executada a partir de outro programa, chamado interpretador.

O *daemon* pode ser desenvolvido em qualquer tipo de linguagem, porém, foi escolhida o *python*, pois é, uma linguagem de simples implementação, o código não se torna muito extenso, além de existir uma grande comunidade de desenvolvedores que auxiliam no suporte de dúvidas.

Neste projeto o *daemon* é utilizado somente para passar as informações, os IDs das *tags*, que chegam na USB onde esta conectada a placa PCA1000, que opera como leitor de RFID. Após receber o ID da *tag*, um programa em PHP que está sendo executado em um servidor, registra uma nova volta dada pelo nadador, por exemplo.

2.7 Apache

O *Apache* é um servidor *web* desenvolvido pela *Apache Software Foundation*. Um servidor *web* é um programa que recebe uma solicitação do tipo *Hypertext Transfer Protocol* (HTTP) vinda de um *browser*, processa as informações através de um aplicação servidora, e responde o resultado do processo para o *browser* (ALBUQUERQUE; MAESTRELLI, 2000).

Este servidor é muito popular, dentre as suas principais características estão: fácil instalação e configuração, suporte a *Secure Sockets Layer* - Protocolo de Camada Socket Segura (SSL), suporte a banco de dados e ser um *software open source*. Além disso, ele pode operar em dife-

²<http://www.async.com.br/projects/pnp/pnp.pdf>. Acessado em 19 de junho de 2014.

rentes plataformas de sistemas operacionais como *Windows* e *Linux* (MELO; NASCIMENTO, 2007).

O fator determinante na escolha do servidor *Apache* no desenvolvimento do MAN foi sua fácil instalação, configuração e o fato de já estar contido no reservatório *Advanced Packing Tool* (APT) do *Linux*. Outro fator importante é o seu suporte às aplicações em PHP, que é a linguagem escolhida para o desenvolvimento da lógica do programa.

2.7.1 PHP

O *Hypertext Processor* (PHP) é uma linguagem criada em 1994 por *Rasmus Lerdorf*. Esta linguagem tem por objetivo a criação de páginas dinâmicas, ou seja, permite interação com o usuário. Esta linguagem é executada dentro de um servidor, sendo enviado, para o *browser* que está fazendo a solicitação, apenas o *Hypertext Markup Language* (HTML) com o resultado do processo solicitado. Desta maneira, é possível interagir com aplicações existentes no servidor, como por exemplo, o banco de dados (ALMEIDA, 2004).

Algumas das características de se utilizar PHP estão a possibilidade de trabalhar com arquivos *Extensible Markup Language* (XML) e a possibilidade de trabalhar com banco de dados do tipo *MySQL* com comandos nativos para a manipulação do mesmo. O PHP não necessita de licenças para utilização, seu modelo de licença é do tipo *General Public Licence* (GPL) (MELO; NASCIMENTO, 2007).

O programa de acompanhamento de treino desenvolvido por *Mário Felipe Raulino* (RAULINO, 2013) foi embarcado em uma placa *Raspberry Pi*, com o objetivo de deixar o dispositivo mais próximo da piscina junto com um monitor, porém, a baixa capacidade de memória RAM junto com a JVM, causava lentidão na operação do sistema.

Este programa não será embarcado em *Raspberry Pi*, o programa será desenvolvido em um computador/servidor, tornando possível o acesso dele por qualquer computador, que se encontre em uma rede de computadores, através de um *browser*,

No MAN o programa em PHP é responsável por tratar a lógica do acompanhamento do treino de natação, portanto, devem ser feitas várias solicitações do processo em PHP até o término do treino. Para que estas solicitações sejam automáticas, foi utilizada a biblioteca *JQuery*.

2.7.2 JQuery

O *JQuery* é uma biblioteca de *JavaScript* que possibilita aumentar o dinamismo e interação com a aplicação *web*. Com esta biblioteca é possível buscar informações no servidor sem a necessidade de recarregar a página, trabalhar com efeitos visuais e animação, modificar apresentações e conteúdos (SILVA, 2008).

Esta biblioteca se torna muito útil para o MAN, pois, enquanto o sistema acompanha o treino são feitas várias solicitações para a aplicação *web* sem a necessidade de recarregar a página. Por exemplo, o programa em *python* recebe a ID da *tag* e repassa para um banco de dados, a cada cem milissegundos o *JQuery* executa o programa em PHP que lê o conteúdo deste banco de dados.

2.8 Banco de Dados

Banco ou base de dados é um conjunto integrado de dados que é utilizado por múltiplos objetivos e manipulado por diferentes utilizadores, de maneira concorrente. Algumas das características fundamentais do banco de dados é a integração, a partilha, a concorrência, os múltiplos objetivos e os múltiplos utilizadores do banco (TAKAI; ITALIANO; FERREIRA, 2005).

SGBD são softwares que possibilitam acesso a um ou mais base de dados, possibilitando a definição, acesso concorrente, manipulação e controle dos dados, assegura integridade, segurança e recuperação do banco de dados (TAKAI; ITALIANO; FERREIRA, 2005).

Existem vários tipos de SGBDs, dentre eles um dos mais populares é o *MySQL*.

2.8.1 MySQL

O *MySQL* é um SGBD muito popular, pois, é um *software open source* e não há necessidade de adquirir licenças para sua utilização. Suporta SQL³ e é um dos bancos de dados relacionais mais utilizados no mundo, com cerca de cinco milhões de ativações (NEVES; RUAS, 2005).

Ele é um banco de dados relacional, ou seja, armazena dados em tabelas separadas em vez de colocar todos os dados em um só local, onde é mantido um vínculo de relação entre elas (MER). O *MySQL* é um banco de dados rápido, confiável e fácil de utilizar. Foi desenvolvido originalmente para lidar com bancos muito grandes de maneira mais rápida que as soluções mais comuns (MYSQL, 2010). Algumas características são: o código fonte ser desenvolvido

³SQL - Linguagem de pesquisa declarativa, padrão para banco de dados relacional

em C e C++, e opera em diferentes plataformas. Possui APIs para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby e Tcl (MYSQL, 2010).

O *MySQL* foi escolhido para ser o banco de dados utilizado pelo MAN, pois não há necessidade de aquisição de licenças, possui suporte para PHP e é um banco de dados relacional.

2.8.2 MER - Modelo Entidade Relacionamento

O MER é um modelo conceitual para implementação de banco de dados, ou seja, a descrição de um banco de dados, independente do SGBD que foi escolhido, com este modelo será possível definir os dados que serão armazenados. Mas o MER não informa como os dados serão armazenados em nível de SGBD (HEUSER, 1998).

O Diagrama Entidade Relacionamento (DER) é a representação gráfica do modelo entidade relacionamento. Com ele é possível descrever de maneira gráfica a relação entre as tabelas e como elas se relacionam, facilitando a visualização e o entendimento.

Para entender o modelo, é necessário entender alguns conceitos, como entidade, atributo e relação.

- Uma *entidade* é a abstração de objetos ou conceitos do mundo real onde se deseja manter informações no banco de dados. No DER é representado por um losango.
- Um *relacionamento* é uma associação entre as entidades e no diagrama é representado por um retângulo.
- *Atributos* são os dados armazenados em cada ocorrência de uma entidade ou relacionamento. São representados por círculos ou elipses.

Outro conceito relevante é a cardinalidade, que é a quantia (mínima ou máxima) de ocorrências de um entidade que pode estar associada a outra entidade através de um relacionamento. Para fins práticos não é necessário distinguir a quantia de cardinalidade máxima maior que 1. Portanto, são consideradas apenas duas cardinalidades a cardinalidade *1* e a cardinalidade *n*, que representa muitos. A cardinalidade máxima pode ser utilizada para definir a classificação das relações, sendo *1:1* (um para um), *1:n* (uma para muitos) e *n:n* (muitos para muitos).

A Figura 2.8 representa um exemplo de uma relação 1:n, onde um curso pode conter muitos (n) alunos inscritos.



Figura 2.8: Exemplo de um Diagrama Entidade Relação (HEUSER, 1998).

No MAN, existem algumas tabelas como as treino (proposto e realizado) e usuário que devem ser mantido um vínculo. O modelo MER é utilizado para associar as tabelas e assim saber de quais treinos (inseridos na tabela de treino realizado) o usuário realizou.

No próximo capítulo é descrito o desenvolvimento do sistema, Começando por uma visão geral do sistema e em seguida cada uma das partes deste, como o *daemon* e a maneira de como ele interage com o programa de acompanhamento de treino. Também são descritos o banco de dados e a maneira de como as tabela se relacionam. E por fim a lógica do programa de acompanhamento de treino.

3 *MAN - Monitoramento de Atividades de Natação*

O MAN é um sistema usado para monitorar treinos de natação em piscinas. Ele utiliza a tecnologia de RFID para acompanhar os treinos dos nadadores. Sua principal característica é a sua capacidade de identificar cada atleta que está nadando na raia e acompanhar o treino de cada um sem que um interfira no outro. Para isso, cada nadador deverá usar pelo menos uma etiqueta de identificação, chamada de *tag* ou etiqueta de RFID, para que um leitor de RFID possa identificá-lo na piscina e assim monitorar o seu treino. O leitor de RFID sempre que receber o ID de uma *tag* repassa-a, através do *middleware*, para um *software* que fará o processamento desta informação. Assim, poderá ser feita a contabilidade das voltas dadas na piscina e ser registrado o tempo de cada volta.

Além de implementar este sistema, o MAN cria uma interface com o usuário, onde é possível planejar os treinos e guardá-los em um banco de dados para serem comparados com treinos realizados, que também são salvos no bancos de dados. Os sistemas de planejamento de acompanhamento de treino são importantes para que o sistema MAN, que faz uso da tecnologia RFID, funcione adequadamente como um equipamento de acompanhamento de treinos de natação.

A Figura 3.1, apresenta um diagrama que descreve o MAN. Nesta figura é apresentado um equipamento que faz a leitura das *tags* de RFID e repassa para o sistema através de uma comunicação serial. Um *daemon*, que é o processo *produtor* de recursos para o sistema, monitora a entrada serial, e registra em uma tabela do banco de dados, a ID da *tag*, o *time stamp* e a data e hora do momento da passagem da *tag* pelo leitor. Esta tabela, de nome *serial*, é utilizada somente para passar os dados que entram pela serial para o *software* em PHP. O programa em PHP é o processo *consumidor* ele lê o último registro deste banco de dados, que contém informações como por exemplo, o valor da *tag* e o *time stamp*¹ do momento da leitura, e salva em outra tabela, que está relacionada com a tabela de treinos realizados. Este programa em PHP

¹O *Time Stamp* é a contagem de tempo em segundos desde 1 de janeiro de 1970.

também é responsável pelo cadastro do atleta, dos treinos propostos e configuração de piscina. No momento da realização do treino também é feito um registro na tabela de treino realizado. Esta última tabela é para relacionar as tabelas de usuário e treino proposto com a tabela dos registros de voltas. As tabelas de treinos propostos e treinos realizados são relacionadas entre si para que o nadador possa comparar o seu desempenho de acordo com a proposta feita.

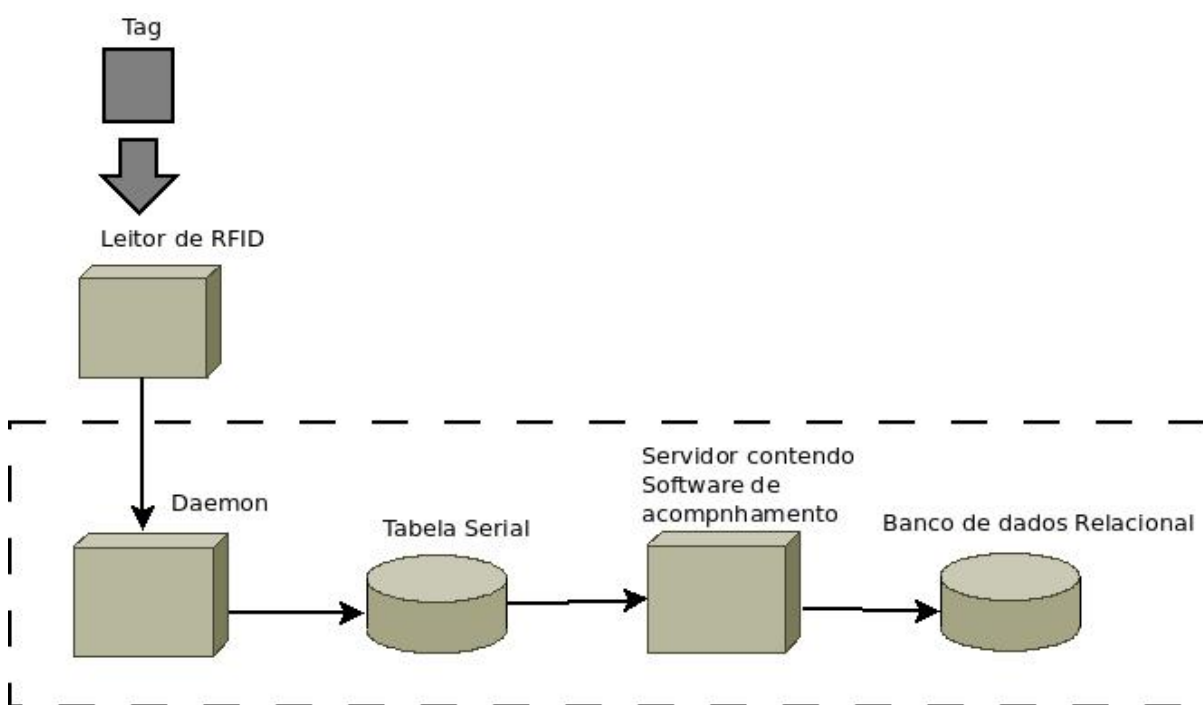


Figura 3.1: Diagrama de blocos do sistema MAN.

3.1 Daemon no MAN

O leitor de RFID ao receber o ID enviado pela *tag* repassa esta informação para o computador através do *middleware*. Neste computador (ou servidor) está rodando um *software* que vai registrar o ID e o *time stamp* da leitura deste no banco de dados. O *software* que faz a interface com o leitor e o registro no banco de dados é chamado de *daemon*. O meio de comunicação entre o leitor e o computador é uma porta serial. O *daemon* é necessário porque o *software* do MAN foi desenvolvido em PHP que não possui uma interface adequada para ficar acompanhando continuamente a chegada de novos conteúdos na porta serial.

Este *daemon*, desenvolvido em *python*, fica em um estado “dormindo” e quando chega alguma informação pela serial, através do leitor, faz o registro na tabela *serial*. Esta tabela é usada somente para “passar” dados entre o *daemon* e o programa de acompanhamento de treino, a mesma não se relaciona com as tabelas de usuário, treino proposto, treino realizado e

de voltas. Os campos desta tabela estão descritos na Tabela 3.1.

Tabela 3.1: Tabela com valores e descrições da tabela utilizada pelo *daemon*.

Nome do Atributo	Tipo	Descrição
<i>id_tag</i>	Auto Increment	Valor da identificação do evento.
<i>tag</i>	Varchar	Conteudo da <i>tag</i> lida.
<i>data</i>	Varchar	Data e hora do momento do evento.
<i>time_stamp</i>	Varchar	<i>Time stamp</i> do momeno do evento.

A lógica de operação de programa pode ser vista na Figura 3.2. É uma lógica simples, onde fica aguardando a chegada de informações pela porta USB/serial e quando chegar, e esta informação conter oito caracteres (quantidade de caracteres do código de identificação da *tag*), serão inseridas na tabela de nome *serial*, com a informação de de *time stamp* e data e hora do momento da leitura.



Figura 3.2: Fluxograma de operação do *daemon*

3.2 O Banco de Dados do MAN

Para que o usuário possa acompanhar a evolução do seu desempenho, deve buscar os resultados obtidos em treinos realizados anteriormente. Em um banco de dados podem ser salvos

os dados de um treino. Devem ser gravados os dados referentes as quantidades de séries de um treino, tempo de cada volta e a velocidade de cada volta. Voltas são definidas como sendo o percurso que o nadador leva para ir até a borda oposta e retornar para a mesma borda e série é um conjunto de voltas que serão nadadas continuamente antes de um intervalo de descanso. Estas informações ficam salvas no banco de dados e através do programa em PHP o usuário pode consultar os treinos realizados.

O trabalho desenvolvido (RAULINO, 2013) não visava salvar os dados do atleta, de treino proposto e nem salvar os valores obtidos durante os treinos. Sendo que antes de realizar o treino o usuário deveria pré-determinar o número de voltas, de séries e o tempo de descanso entre as séries.

Um das propostas deste programa é a utilização de banco de dados para salvar os dados do usuário e dos treinos propostos e realizados por este. Os dados de treinos realizados são importantes para futuramente serem consultados, a fim de analisar o rendimento dentro de janelas de tempo grandes, como por exemplo, a evolução do nadador ao longo dos últimos três meses. Também é utilizada uma tabela para configurar piscina, com as informações da piscina onde o MAN está instalado.

3.2.1 Tabela de Piscina

A tabela de nome *piscina* permite configurar a especificação da piscina e da academia de natação onde o MAN está instalado. Com esta tabela é possível manter salvas as informações sobre as dimensões da piscina.

Outra função para esta tabela é a possibilidade de customizar o MAN para a piscina ou academia onde estiver instalado. Podendo apresentar o nome da academia durante o funcionamento do *software*, por exemplo.

Os atributos e suas definições podem ser vistos na Tabela 3.2. Sabendo o comprimento da piscina e o tempo que foi realizado cada volta é possível descobrir a velocidade de cada volta. A largura de cada raia é importante para saber a quantidade e o posicionamento de cada leitor de RFID.

Segundo as normas da FINA (FALCAO, 2009) as piscinas podem ser divididas em dois tipos que a piscina *olímpica* e a piscina *semiolímpica*. A piscina *olímpica* possui 50 metros de comprimento, 25 metros de largura, 8 raias com larguras de 2.5 metros e uma profundidade de 2 metros. A piscina *semiolímpica* possui 25 metros de comprimento, 20 metros de largura, 8 raias com larguras de 2 metros e uma profundidade de 2 metros.

Tabela 3.2: Valores e descrições da tabela *piscina*.

Nome do atributo	Tipo	Descrição
<i>id_piscina</i>	Auto Increment	Valor da identificação do usuário.
<i>nome</i>	Varchar	Nome da piscina onde o equipamento será instalado.
<i>comprimento</i>	Varchar	Comprimento da piscina.
<i>largura_raia</i>	Varchar	Largura de cada raia da piscina.
<i>nr_raia</i>	Varchar	Quantidade de raias que piscina possui.
<i>nome_academia</i>	Varchar	Nome da academia de natação.

3.2.2 Tabela de Usuário

A tabela *Usuario* é responsável pelas informações pessoais do nadador a fim de conhecer o perfil do nadador. Esta tabela de base de dados está descrita na Tabela 3.3.

Os atributos de *nome_atleta*, *email*, *cpf*, *nascimento*, *registro*, *altura*, *peso* e *sexo* são dados cadastrais, como se o usuário estivesse matriculando-se em uma academia, informações como peso, altura e sexo podem ser usadas para estimar o índice de massa corporal do nadador, e também, a perda calórica em um treino. Os atributos *tag1* e *tag2* são referentes aos códigos das *tags* vinculadas ao usuário.

O atributo *senha* é importante para manter a confidencialidade das informações do nadador. Pois sempre que usuário, propõe, realiza ou consulta um treino é solicitado a senha para descobrir qual o usuário e a *id_usuario* do usuário em questão.

3.2.3 Tabela de Treino Proposto

Na tabela de *treino_proposto* é possível salvar os treinos que o usuário realizará futuramente. Esta tabela carrega o código de identificação da tabela de usuário, assim é possível saber de qual usuário é o treino proposto. Alguns campos desta tabela são não foram usada nos testes realizados, mas estão previstos para aprimoramento do software, tais como: *estilo_treino*, *tipo_treino*, *volta_aquecimento*, *volta_desaquecimento*, *tempo_volta_aquecimento*, *tempo_volta_desaquecimento*.

O atributo *id_usuario*, é o atributo que carrega o código de identificação do usuário, que foi criado na tabela de usuário quando o mesmo foi inserido nela. Este atributo é que mantém um vínculo entre a a tabela de *usuario* e a tabela *treino_proposto*.

Os atributos mais relevantes para a realização de um treino são: *nr_series*, *nr_voltas*, *tp_volta*

Tabela 3.3: Valores e descrições da tabela de *Usuario*.

Nome do atributo	Tipo	Descrição
<i>id_usuario</i>	Auto Increment	Valor da identificação do usuário.
<i>nome_atleta</i>	Varchar	Nome do usuário.
<i>email</i>	Varchar	<i>E-mail</i> do usuário.
<i>senha</i>	Varchar	Senha do usuário, será utilizado para acessar os dados de treino realizado e proposto
<i>cpf</i>	Varchar	CPF do usuário.
<i>nascimento</i>	Varchar	Data de nascimento do usuário.
<i>registro</i>	Varchar	Data de registro do usuário.
<i>altura</i>	Varchar	Altura do usuário.
<i>peso</i>	Varchar	Peso do usuário.
<i>sexo</i>	Varchar	Sexo do usuário.
<i>tag1</i>	Varchar	Primeira <i>Tags</i> cadastradas para o usuário.
<i>tag2</i>	Varchar	Segunda <i>Tags</i> cadastradas para o usuário.

e *intervalo_descanso*. São informações necessárias para definir o treino a ser realizado e o *software* poder realizar o acompanhamento do treino.

Os atributos da tabela *treino_proposto* estão descritos na Tabela 3.4.

3.2.4 Tabela de Treino Realizado e Tabela de Voltas

No momento da realização dos treinos o usuário deve selecionar o treino dentre aqueles que foram programados. Com isso é feito um registro na tabela *treino_realizado*, que carrega os valores da chave primária das tabelas *usuario* e *treino_proposto*, para que as tabelas fiquem vinculadas.

Todas as voltas realizadas serão inseridas na tabela *voltas* e cada inserção carrega o *id_treino_realizado*. A finalidade da tabela *treino_realizado* é vincular as inserções na tabela *voltas* com o usuário que as realizou e o treino proposto para o mesmo.

Além das inserções de Identificação (ID) das tabelas *usuario* e *treino_realizado*, também são inseridos os valores de número de séries, número de voltas por série, tempo de cada volta e o intervalo de descanso, entre cada série, do treino que será realizado. O atributo *data* carrega a informação sobre o dia e hora do treino que foi realizado, esta informação é importante quando for realizar uma futura consulta, pois, sabendo o dia que foi realizado é possível identificar o *id_treino_realizado*, sabendo esta informação é possível selecionar todas as voltas da tabela

Tabela 3.4: Valores e descrições da tabela de treino proposto.

Nome do atributo	Tipo	Descrição
<i>id_treino_prop</i>	Auto Increment	Valor da identificação do treino.
<i>id_usuario</i>	Varchar	Valor de identificação do usuário, no qual o treino está vinculado.
<i>estilo_treino</i>	Varchar	Estilo de nado que será utilizado no treino.
<i>tipo_treino</i>	Varchar	Tipo de treino proposto.
<i>volta_aquecimento</i>	Varchar	Quantidade de voltas a serem realizadas no aquecimento.
<i>tempo_aquecimento</i>	Varchar	Tempo planejado para cada volta de aquecimento.
<i>volta_desaquecimento</i>	Varchar	Quantidade de voltas a serem realizadas no final do treino para “soltar” a musculatura.
<i>tempo_desaquecimento</i>	Varchar	Tempo planejado para cada volta de desaquecimento.
<i>nr_series</i>	Varchar	Número de séries.
<i>nr_voltas</i>	Varchar	Número de voltas a serem realizadas até o próximo intervalo de descanso.
<i>tp_volta</i>	Varchar	Tempo em que cada volta deverá ser realizada.
<i>intervalo_descanso</i>	Varchar	Tempo do descanso entre uma sequência de voltas.

volta que carrega este ID.

As descrições da tabela *treino_realizado* estão descritas na Tabela 3.5. As descrições da tabela *voltas* estão descritas na Tabela 3.6.

Cada inserção na tabela *voltas* salva o valor de *time stamps* válidos e a ID da tabela *treino_realizado*. Os *time stamp* válidos são os valores de tempo das viradas, sendo considerado o *time stamp* de início de série. Este tempo de início de série fica salvo no campo *ts_inicio*, da tabela de treino realizado.

Além de registrar o tempo de início de série, no campo *ts_inicio* também é registrado o *time stamp* de cada volta, pois, para descobrir o tempo da volta é feita a subtração do *time stamp* atual menos o *time stamp* do campo *ts_inicio*, segundo a equação 3.1.

$$tempo_volta = time\ stamp\ atual - ts_inicio \quad (3.1)$$

Tabela 3.5: Valores e descrições da tabela de treino a ser realizado.

Nome do atributo	Tipo	Descrição
<i>id_treino_realizado</i>	Auto Increment	Valor da identificação do treino a ser realizado.
<i>id_treino_prop</i>	Varchar	Valor de identificação do treino proposto, no qual o treino está vinculado.
<i>id_usuario</i>	Varchar	Valor de identificação do usuário, no qual o treino está vinculado.
<i>data</i>	Varchar	Data da realização do treino.
<i>nr_series</i>	Varchar	Número de séries a serem realizadas no treino.
<i>nr_voltas</i>	Varchar	Número de voltas a serem realizadas até o próximo intervalo de descanso.
<i>tp_volta</i>	Varchar	Tempo em que cada volta deverá ser realizada.
<i>intervalo_descanso</i>	Varchar	Tempo do descanso entre uma sequência de voltas.
<i>ts_inicio</i>	Varchar	<i>Time stamp</i> inicial de cada série, este valor é alterado sempre que começa uma série.

Para descobrir a velocidade é feito a divisão do comprimento da piscina (definida na tabela da piscina) e o tempo da volta em segundos. Sendo assim a velocidade é em metros por segundos, segundo a equação 3.2 (VILLATE, 2012).

$$velocidade = \frac{comprimento}{tempo_volta} \quad (3.2)$$

3.2.5 Diagrama de Entidade Relação do MAN

Para que as tabelas citadas anteriormente possam ser vinculadas entre si, foi criado um DER entre elas para que possam ser vinculadas cada vez que ocorre a inserção em uma tabela. A Figura 3.3 representa o diagrama entidade relação do sistema MAN. Este diagrama foi criado com o auxílio do programa *MySQL Workbench*, um programa utilizado para construção de DERs.

Com ela é possível fazer a relação entre as tabelas de *usuario*, *treino_proposto*, *treino_realizado* e *voltas*. Cada tabela possui um campo de ID, sempre que é feita uma nova inserção em cada tabela esse ID é auto-incrementado, ou seja, a cada inserção é criado automaticamente um código neste campo de identificação.

Tabela 3.6: Valores e descrições da tabela *voltas*.

Nome do atributo	Tipo	Descrição
<i>id_volta</i>	Auto Increment	Valor da identificação do treino a ser realizado.
<i>id_treino_realizado</i>	Varchar	Valor de identificação do treino realizado, no qual a volta está vinculado.
<i>tag</i>	Varchar	Valor da <i>tag</i> lida.
<i>time_stamp</i>	Varchar	<i>Time stamp</i> do momento da leitura da <i>tag</i> .
<i>tempo_volta</i>	Varchar	Valor obtido através da subtração do <i>time stamp</i> da última leitura com o <i>time stamp</i> desta leitura.
<i>velocidade</i>	Varchar	Velocidade desta volta.

A relação entre as tabelas se dá através destes códigos de ID, cada inserção na tabela possui um ID próprio e também carrega o ID de outra tabela. Com isso é possível fazer um conexão entre as tabelas com o ID pertencente da outra, por exemplo, uma inserção na tabela de *treino_proposto* carrega o *id_usuario*, com isso é possível ter um vínculo entre um usuário e um treino proposto.

Quando um atleta realiza o seu treino, ele deve selecionar o treino desejado entre os treinos propostos para o mesmo. Quando é selecionado um treino, é feita uma inserção na tabela de treino realizado. Esta inserção gera um código de identificação de treino proposto (*id_treino_prop*) e insere o código de identificação do nadador (*id_usuario*), mantendo um vínculo entre o usuário, treino proposto e o treino a ser realizado. Estas informações serão importantes quando se desejar comparar os treinos propostos com os realizados ao longo de uma janela de tempo.

Quando o treino é inicializado e as voltas começam a serem contabilizadas, são feitas inserções na tabela *voltas*, cada inserção carrega o código de ID de treino realizado (*id_treino_realizado*). Com isso é possível saber para qual treino realizado as voltas contabilizadas estão marcadas. E como a tabela *treino_realizado* carrega o *id_usuario* é possível saber para qual usuário a volta pertence.

3.3 Interface com o Usuário

O nadador necessita de uma interface gráfica onde possa fazer o seu cadastro de usuário do sistema, cadastro e realização de treino e também possa consultar o treinos que foram reali-

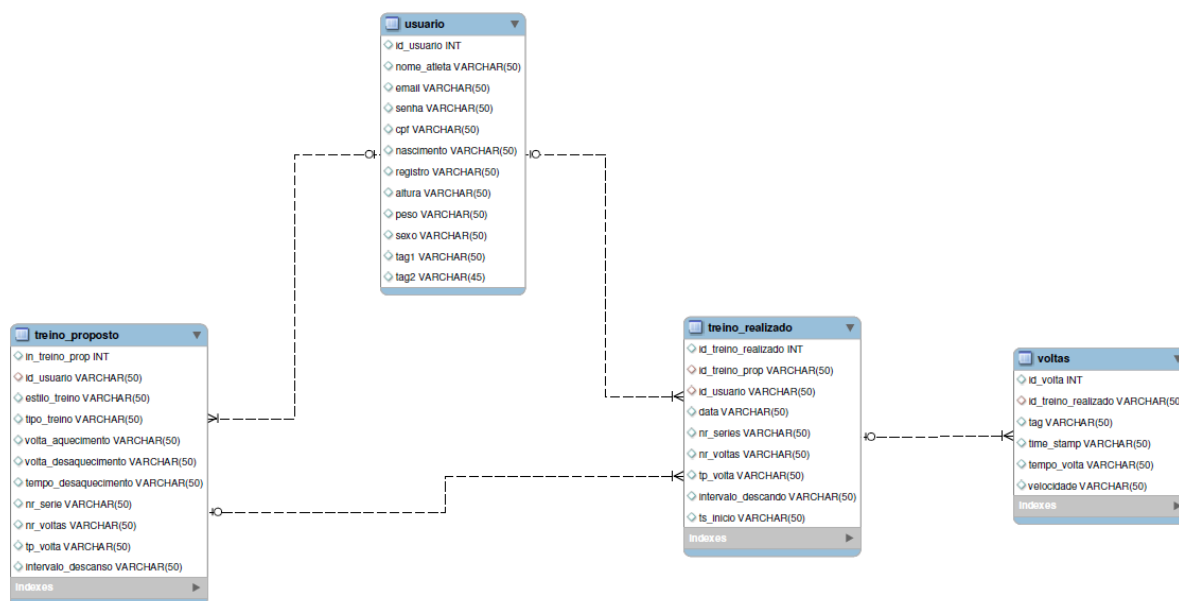


Figura 3.3: Diagrama entidade relação do MAN, criado com auxílio do programa *MySQL Workbench*

zados. O diagrama da Figura 3.3 ilustra a lógica associada a interface que alimenta as tabelas citadas na seção anterior.

A interface, bem como a lógica do programa, foi desenvolvido em PHP, e está sendo executada dentro de um servidor *Apache*. Para acessar basta abrir um *browser* e digitar o endereço *http://(ip do servidor)/MAN/index.php*.

Nesta interface, também há uma tela para configuração do banco de dados, definindo o endereço de *Internet Protocol* - Protocolo de *Internet* (IP), usuário e senha para acesso do banco de dados, estas informações são necessárias para realizar inserções e consultas no banco de dados.

3.3.1 Configuração do Sistema

Na tela de configuração do sistema é possível definir usuário, senha e endereço IP do banco de dados. Estes dados são importantes para fazer a manipulação do banco de dados (consulta e inserção).

Os dados de configuração do sistema ficam salvos em um arquivo do tipo XML. Foi escolhido este tipo de arquivo devido a facilidade de separação do conteúdo para formatação. Para realizar a leitura e escrita deste arquivo é necessário buscar somente os campos especificados, enquanto em um arquivo de texto comum é necessário contar linhas e caracteres para poder editar o mesmo.

3.3.2 Cadastro de Usuário

Na tela de cadastro de usuário, como o nome já diz, é possível realizar o cadastro do usuário, estes dados são inseridos na tabela *usuario*.

Os campos são preenchidos conforme a Tabela 3.3, sendo que os campos de *tag1* e *tag2* são lidas através das informações vindas pelo leitor de RFID ou manualmente.

Quando ocorre o cadastro de usuário na tabela *usuario* as informações mais importantes para serem inseridas na tabela são: o *nome do usuário*, a senha e os valores de *tag1* e *tag2*.

É importante ter a informação da senha, pois cada vez que algum usuário for criar seu treino ou realizar o mesmo, será solicitado a senha para saber de qual usuário se trata, e inserir nas tabelas do banco de dados a identificação do usuário correto. Sempre que é feito o cadastro de usuário o sistema verifica se já há outra senha igual na tabela, portanto, cada senha inserida na tabela *usuario* é diferente e não terá a mesma senha para dois usuários.

3.3.3 Cadastro de Treino

Para acessar a tela de cadastro de treino é solicitado ao usuário a sua senha, assim é possível consultar a tabela de *usuario* a chave primária do mesmo ao qual a senha esta associada. Esta informação será inserida na tabela de *treino_proposto*, conforme Tabela 3.4.

Os campos mais relevantes da tabela de treino proposto, além do *id_treino_prop* que é auto-incrementado, são os campos de *nr_serires*, *nr_voltas*, *tp_volta* e *intervalo_descanso*. Os outros campos são somente para nível de cadastro e futuras aplicações do sistema MAN.

3.3.4 Cadastro de Piscina

Como o MAN fica instalado em uma piscina, esta tela de configuração faz o cadastro desta piscina. Esta tabela possui somente uma inserção e sempre que é alterado os dados desta piscina é feita uma atualização (*update*) no registro já existente na tabela de banco de dados, conforme a Tabela 3.2.

O dado mais importante desta tabela é o referente ao comprimento da piscina. O restante é apenas para nível de cadastro e as possibilidades de futuras implementações no sistema MAN.

3.3.5 Realização do Treino

Quando é acessado a tela para realização de treino, a primeira coisa que é solicitada ao usuário é a senha do mesmo. Com esta senha é possível acessar os dados do usuário e uma lista com os treinos propostos para o mesmo. O usuário deve selecionar o treino proposto que deseja realizar.

Ao selecionar o treino é feita uma inserção na tabela de treino realizado. Após esta inserção começa o processo para acompanhamento do treino. Cada vez que uma volta é contabilizada, é feita uma inserção dentro da tabela *voltas*. Cada inserção feita é preenchida com o código de identificação da tabela de treino realizado (*id_treino_realizado*), com isso é possível saber qual volta esta vinculada com qual treino realizado.

O processo de acompanhamento do treino, é um *script* feito em *JQuery* que a cada cem milissegundos executa um programa em PHP, que executa a lógica do programa. A lógica do programa está descrita no fluxograma da Figura 3.4.

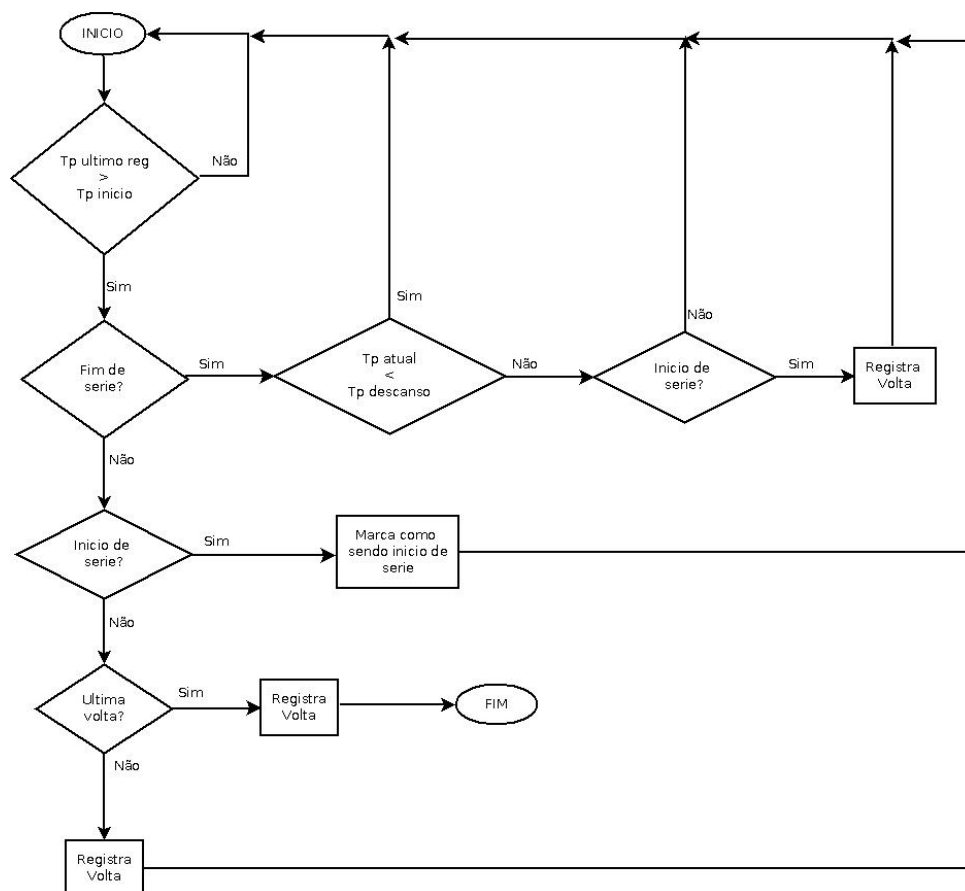


Figura 3.4: Diagrama de blocos do programa de acompanhamento de treinos.

O primeiro passo deste programa em PHP é selecionar os campos da tabela *serial*, que contém os dados recebidos pelo leitor de RFID.

Após selecionar estes dados da tabela *serial* (valor da *tag* e *time stamp* da leitura), começa a realizar o processo para registrar a volta na tabela de *voltas*.

O primeiro passo do processo PHP é analisar se o *time stamp* do último registro da tabela *serial* (*Tp Ultimo Reg*) for maior que o *time stamp* do início do treino (*Tp inicio*), assim irá contabilizar somente os registros após o treino ser inicializado.

Quando ocorre a leitura de *tag* o sistema analisa três situações possíveis que são: fim de série, início de série e última volta.

A primeira situação a ser analisada, é se for fim de série. Neste caso é analisado se a variável *Tp atual* (que é a diferença entre o *time stamp* atual e o *time stamp* da última volta registrada na tabela *voltas*) é menor que o tempo de descanso (*Tp descanso*) definido para este treino. Caso for maior é considerado que está em tempo de descanso e aguarda até que a variável *tempo atual* seja maior que o tempo de descanso, significando que saiu do tempo de descanso e irá começar uma próxima série.

Caso seja início de série, é registrado a primeira leitura, de abertura de volta, somente a partir da segunda leitura após início de série que é contabilizada na tabela *voltas*, este Ponto de decisão é importante para marcar o primeiro leitura no leitor de RFID.

Caso seja última volta do treino, ou seja, se a quantidade de registros na tabela *voltas* com o *id.treino_realizado* for igual ao número total de voltas do treino proposto menos um. Se estiver nesta última volta, então, é feito o registro na tabela *voltas* e sai do processo de acompanhamento de treino e termina o treino.

Caso não seja nenhuma das três situações, a leitura da *tag* é feita uma inserção na tabela *voltas*, contabilizando uma volta completada.

Alguns casos citados são importantes para serem analisados a parte, como o fim de série, o início de série e o procedimento de registro na tabela.

Fim de Série

Uma série é um conjunto de voltas realizadas entre o períodos de descanso. Cada volta completa gera uma inserção na tabela *voltas*, o sistema necessita contar o número de voltas e analisar se é fim de série para entrar no rotina de intervalo de descanso.

O programa de acompanhamento em PHP necessita que seja executado a cada intervalo de tempo, para que consulte o ultimo registro na tabela *voltas*. O *JQuery* executa este programa a cada intervalo de cem milissegundos, foi definido que seja neste intervalo de tempo para que

não ocorra um *delay* no momento de carregar as informações na tela e esta visualização tenha seja mais agradável para o usuário.

Como a cada cem milissegundos o *JQuery* executa o programa em PHP, não é possível ter vetores (no programa em PHP) para contabilizar a quantia de voltas já realizadas. Pois cada vez que executa o programa em PHP este vetor iniciaria com valor zero, e não seria possível contabilizar a quantia de voltas já realizadas.

Para contabilizar a quantidade de voltas já realizadas é contabilizado o número de inserções com o *id_treino_realizado* na tabela *voltas*.

Para saber se é fim de série, é utilizado o resto da divisão entre a quantidade de voltas já realizadas e o número de voltas que há em uma série, conforme equação 3.3. Se este resto for igual zero significa que a série chegou ao fim. E assim pode entrar na condição para analisar o tempo de descanso.

$$Fim_serie = \frac{quantia_voltas_realizadas}{voltas_serie} \quad (3.3)$$

Início de Série

Dentro da tabela *treino_realizado* existe um campo denominado de *ts_inicio*. Sempre que é feita um inserção dentro da tabela de *voltas*, também é feita uma inserção dentro do campo *ts_inicio* da tabela *treino_realizado*. Este campo irá conter o *time stamp* do momento da volta completada.

Quando chega na condição de fim de série o campo *ts_inicio* é preenchido com espaço vazio. Para considerar início de série o campo *ts_inicio* deve estar vazio. Isso ocorrerá quando passar pela condição de fim de série ou no início do treino, quando o campo está vazio.

Esta condição foi criada, pois o primeiro *time stamp* é somente para abertura de série, para que a volta seja inserida na tabela *voltas*, uma volta deve ser completada.

Registro de Voltas

Sempre que o sistema faz alguma inserção na tabela *voltas* deve ser analisado se as *tags* lidas são correspondentes ao nadador, se o tempo do último registro na tabela *voltas* é diferente do tempo do último registro da tabela *serial* e se já passou dez segundos do ultimo registro na tabela *voltas*. O fluxograma destas é apresentado na Figura 3.5.

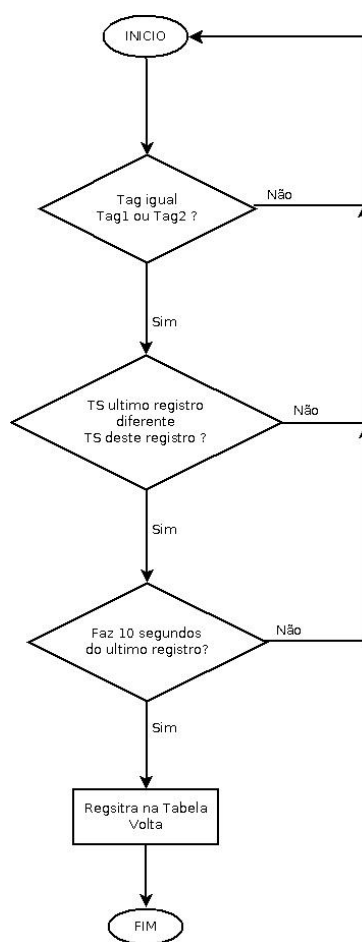


Figura 3.5: Diagrama de blocos do processo de registro na tabela *voltas*.

A primeira análise a se feita é identificar se a *tag* recebida correspondente ao nadador. Se não for é porque o conteúdo é lixo e será descartado. Caso contrário, o nadador foi identificado.

A próxima análise verifica se o *time stamp* da última volta registrada na tabela *voltas* é diferente do *time stamp* do último registro da tabela *serial*. É realizada esta análise para o programa de acompanhamento não ler a mesmo registro na tabela *serial* e contabilizar como volta válida.

Em seguida o programa faz um terceira análise, referente a diferença dos tempos de inserção. O programa prevê que a diferença entre a inserção de registros na tabela *voltas*, deve ter no mínimo dez segundos esta diferença. Este tempo foi previsto para que o nadador saia do raio de alcance do leitor de RFID, este tempo pode ser alterado.

Depois que todas estas análises foram realizadas é feita a a volta é inserida na tabela *voltas*, conforme Tabela 3.6.

Após completar todas as voltas do treino, o treino é encerrado e o sistema de acompanhamento não registra mais nenhuma volta.

Neste capítulo foram descritos o procedimento de construção do sistema MAN. No próximo capítulo descreve os testes e as validações realizadas para comprovar o funcionamento do MAN.

4 Testes

Neste projeto foi relatado o procedimento para o desenvolvimento do MAN, um sistema para acompanhamento de atividades de natação de forma remota, que registra a contagem de voltas, séries e o tempo de cada volta. Para o desenvolvimento deste sistema foi utilizado a tecnologia RFID para marcar o tempo de cada volta, e um *software* em PHP faz o acompanhamento do treino, contando as voltas.

Este capítulo relata os testes realizados para comprovar o funcionamento deste sistema. Com os resultados obtidos foi possível observar seu funcionamento e algumas limitações na operação do sistema que serão descritas na Sessão 4.6.

Os primeiros testes foram realizados em laboratório para analisar o comportamento do *kit* de desenvolvimento nRF51, da fabricante *Nordic*, com o *daemon*. Nestes testes foram avaliadas as chegadas dos Ids das *tags* até o *daemon* e o registro destas no banco de dados.

Em seguida foram realizados os testes de campo. Os primeiros testes realizados, em campo, foi de alcance e a atividade foi variação de ganho de potência, para que consigamos um alcance aceitável entre o leitor e a *tag*. Estes testes foram necessários para que definir uma distância ótima entre a posicionamento do leitor de RFID e passagem do nadador próximo ao leitor, que fique de uma forma que não ocorra erros de leitura e nem que leia a informação da *tag* antes do nadador chegar próximo a borda da piscina.

Para avaliar desempenho do software de acompanhamento, foram criados treinos para validar o funcionamento e verificar se o tempo das voltas correspondem com os tempos aproximados realizados através de um cronômetro.

4.1 Informações Básicas Para Realização dos Testes

O *kit* de desenvolvimento da *Nordic* foi utilizado como o sistema de RFID para o MAN. Este sistema opera em uma faixa de frequência de 2.4GHz e possui uma potência máxima de 4 dBm, podendo variar até -20 dBm, este ganho pode ser alterado através de um *software*

desenvolvido em C.

Este *kit* é composto por três placas que são transceptores, a PCA10000 que para o MAN foi utilizada como leitor de RFID e deve ficar situado em uma das bordas da piscina, as placas PCA10004 e PCA10005 foram utilizadas como *tags*, conforme descrito no Capítulo 3, uma deve ficar no punho e outra deve ficar no tornozelo do nadador.

Os primeiros testes realizados foram em laboratório, para analisar o comportamento do MAN. Em seguida foram realizados testes em campo, primeiramente para definir a área de cobertura da leitura das *tags*. Em seguida testes para analisar o comportamento do MAN durante um treino de natação.

4.2 Teste de Laboratório

Nos testes realizado em laboratório para analisar o comportamento do *daemon* e *kit* de desenvolvimento da *Nordic*, onde o objetivo era verificar se o *daemon* conseguia ler a ID das *tag* e repassar para a tabela *serial*, no banco de dados, sem apresentar erros.

Nestes testes de laboratório um programa em *python* foi editado para que, além de inserir as informações de ID da *tag*, *time stamp* e data e hora da leitura, também exiba as mesmas em uma tela. Foi acompanhado a exibição dos dados na tela e a inserção dos mesmos na tabela *serial* e as mesmas não apresentaram erros de inserção.

Este teste foi realizado por aproximadamente 30 minutos, forma utilizadas duas *tags*, uma sendo a placa PCA10004 e a outra sendo a placa PCA10005, com uma alcance de dois metros entre as placas e os leitores e a potência definida foi de -4dBm. Uma *tag* possuía o ID: A0ABCDEF e outra possuía o ID: B0ABCDEF, e o *daemon* registrou as duas *tags*, no banco de dados. Nestes testes não foram realizados testes de alcance.

4.3 Testes de Campo

Os testes de campo foram realizados na piscina do CEFID. O Centro de Ciências da Saúde e do Esporte (CEFID) é um campus da Universidade do Estado de Santa Catarina (UDESC) que desenvolve atividades de pesquisa, ensino e extensão nas áreas da Saúde e do Esporte. A piscina no qual forma realizados os testes é uma piscina com 25 metros de comprimento.

No primeiro teste realizado, com uma potência de -4 dBm, o dispositivo utilizado foi o PCA10005, dispositivo que possui uma antena externa, no pulso. O dispositivo foi sendo dis-

tanciado do leitor de RFID (PCA10000), enquanto o programa em *python* exibia na tela do computador a leitura do dispositivo, neste teste o alcance ultrapassou o comprimento da piscina de 25 metros. Este procedimento pode ser visualizado na Figura 4.1, onde a leitura da *tag* é exibida em uma terminal. Nestes testes eram considerado um tempo de 10 segundos antes do registro da volta, considerando que neste período o nadador pudesse sair do raio de cobertura do leitor, porém, como a potência era alta, a área de cobertura era grande, fazendo com que a volta era contabilizada antes de ser concluída ou ainda na metade da piscina. A utilização deste tempo foi descartado e a potência foi calibrada até que o alcance entre o leitor e a *tag* estejam em uma distância aceitável para que reconheça somente a leitura no momento da virada.



Figura 4.1: Leitura da *tag* sendo exibida em um terminal.

Devido o alcance da leitura da *tag* ultrapassar toda a extensão da piscina, a potência da *tag* foi alterada para -12 dBm, e o alcance chegou na metade piscina, aproximadamente 12.5 metros, em seguida foi feito o mesmo teste com a placa PCA10004 (dispositivo que possui antena impressa) sendo utilizada como *tag* no pulso e a diferença entre o alcance das duas *tags* chegou a 1.25 m de diferença. Desta maneira o sistema não funcionaria corretamente, pois quando o nadador chegava na metade da piscina considerava como virada e registrava como volta completada. Neste testes as *tags* forma utilizadas no pulso.

Nestes testes foram realizados uma série de nados, somente para analisar o área de cobertura do leitor de RFID. Foram realizados nados no estilo *crawl* e nados no estilo peito. Quando a *tag* estava no pulso, com potência de -12dBm, e foi realizado o nado estilo *crawl* a área de leitura

da *tag* era na metade da piscina. Quando foi realizado os testes com nado peito, utilizando os mesmos parâmetros do nado *crawl*, não foi lida a ID das *tags* em nenhum momento, pois nesse estilo de nado, o braço sempre era mantido submerso. Porém, isso é uma característica do nadador, pois alguns retiram o punho da água durante o nado estilo peito, o que ocasionaria uma leitura caso estivesse na área de abrangência de cobertura do leitor.

Por último o ganho foi alterado para -20dBm, a menor potência possível. Com este ganho, o sinal era reconhecido com aproximadamente 1 metro de distância do leitor de RFID. Porém, com este ganho a virada do nadador é sempre muito próxima a borda da piscina, tão próxima que o nadador toca a borda da piscina. Neste ultimo teste foi utilizada a placa PCA10005 na touca, a Figura 4.2 mostra a *tag* sendo usada na touca e o leitor sendo posicionado muito próximo a área da chegada do nadador.



Figura 4.2: Foto exibindo o treino, com potência de -20dBm e a *tag* na touca.

Durante os testes de alcance ocorreram dois momentos que não foi possível realizar a leitura das *tags*, quando estavam submersas e quando as mesmas eram retiradas da água e ficavam envoltas por água por alguns instantes, isso ocorre devido ao fato de ondas de alta frequência não possuírem penetração na água (MIGUENS, 2000). Pelo fato da *tag* possuir um bom alcance foi decidido por na touca do nadador, ao invés de utilizar no pulso ou tornozelo. Na touca foi utilizado a placa PCA10005.

A Tabela 4.1, mostra os ganhos utilizados, as placas testadas e o alcance de cada uma.

Tabela 4.1: Valores dos testes de alcance.

Potência	PCA10004	PCA 10005
-4 dBm	não foi testada	25 metros
-12 dBm	aproximadamente 12.5 metros	aproximadamente 12.5 metros
-20 dBm	não foi testada	1 metro

4.4 Teste de Validação do Daemon

Antes dos testes começarem serem realizados, foram apagadas todas as inserções da tabela *serial*, conforme Tabela 3.6. A Figura 4.3, apresenta algumas inserções na tabela *serial*, realizadas durante o teste de campo. Na primeira coluna desta imagem, apresenta o ID da inserção na tabela, a segunda coluna apresentando o código de identificação da *tag* (B0ABCD), a terceira coluna informa data e hora do momento da inserção e a quarta o *time stamp* do momento da inserção. O valor 72549 no final da tabela é a quantidade de inserções na tabela *serial*.

O *daemon* insere na tabela *serial* para que nenhuma leitura seja perdida. A cada cem milissegundos o programa em PHP busca a última informação inserida na tabela *serial*, a fim de poder realizar a rotina de acompanhamento. O próximo teste a ser realizado é para avaliar o programa de acompanhamento.

4.5 Teste de Acompanhamento de Treino

Os primeiros testes de acompanhamento apresentaram erros na leitura, pois o sistema despreza leituras de *tags* feitas no intervalo de dez segundos após a primeira leitura, como nos primeiros testes de potência o alcance era grande, a volta era registrada na metade da primeira volta ou muito antes de chegar na borda da piscina. Isso não ocorreu quando a potência do sinal passou para -20dBm, pois o alcance se restringiu a 1 metro.

Após achar um alcance ideal para operação do sistema de RFID, definir que a *tag* deve ficar na touca do nadador e comprovar que o *daemon* esta operando corretamente foi realizado um teste para comprovar o funcionamento do programa de acompanhamento.

Foi criado um usuário na tabela *usuario* com o código de *tag* sendo B0ABCDEF. Foi utilizado a PCA10005 como *tag*, posicionada na touca do nadador e a potência da mesma sendo -20dBm.

E foi proposto um treino com os seguintes parâmetros:

Nr da Inserção	ID da Tag	Data e Hora da inserção na tabela serial	Time Stamp da inserção na tabela serial
72518	B0ABCDEF	2014-08-07 19:00:26	1407448826.52812
72519	B0ABCDEF	2014-08-07 19:00:26	1407448826.58383
72520	B0ABCDEF	2014-08-07 19:00:26	1407448826.63957
72521	B0ABCDEF	2014-08-07 19:00:26	1407448826.68419
72522	B0ABCDEF	2014-08-07 19:00:26	1407448826.72882
72523	B0ABCDEF	2014-08-07 19:00:26	1407448826.77341
72524	B0ABCDEF	2014-08-07 19:00:27	1407448827.09864
72525	B0ABCDEF	2014-08-07 19:00:27	1407448827.29984
72526	B0ABCDEF	2014-08-07 19:00:27	1407448827.4448
72527	B0ABCDEF	2014-08-07 19:00:27	1407448827.52609
72528	B0ABCDEF	2014-08-07 19:00:27	1407448827.5899
72529	B0ABCDEF	2014-08-07 19:00:27	1407448827.63453
72530	B0ABCDEF	2014-08-07 19:00:27	1407448827.67914
72531	B0ABCDEF	2014-08-07 19:00:27	1407448827.72376
72532	B0ABCDEF	2014-08-07 19:00:27	1407448827.76839
72533	B0ABCDEF	2014-08-07 19:00:27	1407448827.81301
72534	B0ABCDEF	2014-08-07 19:00:27	1407448827.85758
72535	B0ABCDEF	2014-08-07 19:00:27	1407448827.90226
72536	B0ABCDEF	2014-08-07 19:00:27	1407448827.95826
72537	B0ABCDEF	2014-08-07 19:00:28	1407448828.01399
72538	B0ABCDEF	2014-08-07 19:00:28	1407448828.09199
72539	B0ABCDEF	2014-08-07 19:01:14	1407448874.97499
72540	B0ABCDEF	2014-08-07 19:01:15	1407448875.04045
72541	B0ABCDEF	2014-08-07 19:01:15	1407448875.31399
72542	B0ABCDEF	2014-08-07 19:01:15	1407448875.37482
72543	B0ABCDEF	2014-08-07 19:01:15	1407448875.4192
72544	B0ABCDEF	2014-08-07 19:02:14	1407448934.2577
72545	B0ABCDEF	2014-08-07 19:02:15	1407448935.08277
72546	B0ABCDEF	2014-08-07 19:02:15	1407448935.27073
72547	B0ABCDEF	2014-08-07 19:03:14	1407448994.36548
72548	B0ABCDEF	2014-08-07 19:03:14	1407448994.44048
72549	B0ABCDEF	2014-08-07 19:03:15	1407448995.04251

+-----+-----+-----+-----+
72549 rows in set (0.16 sec)

Figura 4.3: Resultado das inserções na tabela *serial*.

- Número de séries: 1
- Número de voltas: 3

Como a piscina possui comprimento de 25 metros, e uma volta é completada somente quando faz a virada na borda onde está o leitor de RFID, ou seja, cada volta terá uma distância de 50 metros.

O objetivo deste teste era simular um treino real, propondo um treino para o usuário e coletar os dados deste treino, comparando os resultados obtidos pelo MAN com os resultados registrados em um cronômetro.

A Tabela 4.2 mostra os valores coletados durante o treino realizado. Esta tabela apresenta os valores tempo de cada volta coletada pelo MAN, o tempo medido por um cronômetro e a velocidade de cada volta, de acordo com o que foi calculado pelo MAN. Esta tabela também possui a informação de desvio de valores, que é diferença entre o tempo coletado pelo MAN e o tempo realizado pelo cronômetro. A Figura 4.4 ilustra este teste, onde foi feito o acompanhamento do treino com o MAN e com um cronômetro.

O teste foi realizado da seguinte maneira, quando o nadador começou a nadar foi iniciado a contagem no cronômetro, também sendo iniciado o treino no MAN. Sempre que completava uma volta o MAN reconhecia automaticamente esta volta, enquanto era necessário recomençar a contagem no cronômetro, esse processo pode gerar erro de leitura, já que é manual. O cronômetro foi usado para verificar discrepâncias nas medidas realizadas pelo MAN, porem estas divergências forma minimas.

Tabela 4.2: Valores de acompanhamneto de treino.

Volta	MAN	Cronômetro	Desvio dos Valores	Velocidade calculada pelo MAN
Volta1	58.087s	58.19s	-0.11s	0.8308 m/s
Volta2	58.882s	58.50s	0.38s	0.8491m/s
Volta3	60.182s	60.15s	0.03s	0.8308m/s

Com os testes realizados foi possível analisar algumas limitações do sistema, que serão descritos na próxima sessão, sendo assim possível tirar algumas conclusões e sugerir algumas implementações para futuros trabalhos, que serão descritos nos capítulos seguintes.



Figura 4.4: Teste de acompanhamento do treino de natação.

4.6 Limitações

Após ser submetido a testes de campo, o MAN teve um bom desempenho, porém foram apresentadas algumas limitações quanto a sua funcionalidade.

Uma grande limitação foi referente a emissão do sinal enquanto as *tags* estavam em baixo da água. Sempre que o nadador mantinha as *tags* sob a água os mesmos paravam de enviar sinais. Estes problema não ocorria quando o sistema de RFID era de baixa frequência (125KHz) mas ocorreu quando sistema passou a operar com um sistema de RFID de alta frequência (2.4GHz) e *tags* ativas, pois sinais de baixa frequência podem ser penetrados na água e sinais de alta frequência não possuem esta penetração (MIGUENS, 2000). Para que não ocorra o risco do nadador fazer a virada mantendo as *tags* submersas, foi mantida somente uma *tag* na touca do nadador, única parte que fica fora da água durante o nado.

Outra limitação do sistema é quanto ao início do acompanhamento do treino. Nos testes realizados sempre que o treino era inicializado a pessoa que estava operando o computador de testes deveria dar a partida no programa para que pudesse começar a acompanhar os treinos.

Também ocorreu uma limitação devido ao fato de o programa em PHP desprezar leituras de até dez segundos após uma leitura de ID de *tag*. Esta rotina foi criada, inicialmente, imaginando um tempo em que o nadador saia da área de alcance do leitor. Porém, quando a potência do leitor é alta e área de irradiação do mesmo se torna muito grande, o programa em PHP registrava a volta na metade da piscina ou muito antes do nadador se aproximar do leitor. Esta limitação não ocorreu quando a potencia das *tags* passou para -20dBm e o sinal de irradiação ficou curto.

Baseado nos testes realizados e nas limitações analisadas foram realizadas conclusões finais do trabalho e também sugerido melhorias para trabalhos futuros, que serão descritos nos próximos capítulos.

5 *Conclusões*

O Monitoramento de Atividades de Natação (MAN), é um sistema para acompanhar atividades de natação, utilizando o sistema RFID, onde foi utilizado o *kit* de desenvolvimento da *Nordic*, que opera com faixa de frequência de 2.4GHz. Foram realizados testes para comprovar o funcionamento, onde foi analisado o comportamento do sistema de RFID com faixa de frequência de 2.4GHz, para atividades de natação. Também foi testado se o *software* poderia contabilizar as voltas e registrar o tempo de cada uma corretamente. Baseado nos testes realizados foram obtidas algumas conclusões.

O sistema MAN junto com o *kit* de desenvolvimento da *Nordic*, se mostrou uma solução eficaz para o acompanhamento de treinos de natação e a frequência de 2.4GHz se mostrou adequada para a implantação do MAN, o *kit* de desenvolvimento nRF51, possui placas operando como *tags* com dimensões pequenas, minimizando a interferência no nado.

Uma grande vantagem das placas do *kit* de desenvolvimento da *Nordic* é a possibilidade de configurar o ganho da potência. Com ela é possível calibrar a posição do leitor de RFID com o raio de alcance de leitura das *tags*. Nos testes realizados foi concluído que com um ganho -20 dBm o raio de leitura é aproximadamente um metro, porém, pode deixar um pouco limitado a leitura caso nadador venha a realizar a virada antes deste raio.

O sistema MAN também se comportou como o previsto, nos testes realizados, o *daemon* fez a leitura das *tags* corretamente e o *software* de acompanhamento operou corretamente apresentando, os tempos de voltas coletadas pelo MAN e os tempos cronometrados apresentaram alguns milissegundos de diferença, o cronômetro pode apresentar erros de leitura, porém foi a fonte de comparação usada para verificar alguma possível discrepância nas medidas realizadas pelo *software* de acompanhamento.

Para que não ocorra o risco do nadador fazer a virada mantendo as *tags* junto da água, e perder a leitura do sinal, foi mantida somente uma *tag* na touca do nadador. Como pode ser ajustado o ganho de potência, colocar a *tag* na touca pode ser uma solução ótima. Assim, a leitura sempre acontecerá e não ocorre interferência no desempenho e no nado do atleta por

usar um equipamento no punho e no tornozelo.

De modo geral o sistema de RFID utilizando alta frequência (2.4GHz), se torna um sistema viável para desenvolvimento de sistemas de monitoramento de atividades de natação. Possui a possibilidade de alterar a potência, podendo alterar a área de cobertura do sinal. A potência recebida pode ser repassada ao *daemon*. Assim, o banco de dados pode ter as informações da leitura da virada contendo também a potência recebida. A potência mais alta é a mais provável de ser a mais próxima do leitor e conseqüentemente da borda, então esta leitura pode ser utilizada para registrar a volta.

Uma outra alternativa para futuras implementações, seria o desenvolvimento do MAN utilizando tecnologia RFID de média potência (com sistemas de 10 a 15MHz). Com esta tecnologia pode alcançar uma média distância, nem muito curta como a de baixa e nem muito longa como a de alta frequência. Além disso sistemas de RFID de média frequência costumam ser de baixo custo.

Outro ponto positivo desta solução, com alta frequência, é a miniaturização de dispositivos de RFID que operam em alta frequência. Isso pode facilitar a aceitação desta solução por parte das academias de natação.

Porém, os testes realizados para comprovar a operação do sistema MAN, não foram testes fidedignos, ou seja, estes serviram apenas para validar a solução diante de possíveis erros de registro de voltas e medidas de tempo. Para que a solução seja considerada robusta, outros testes devem ser realizados, como testes de exaustão.

Baseado nos testes realizados e nas conclusões observadas foram propostas algumas melhorias para trabalhos futuros.

5.1 Trabalhos Futuros

Este trabalho descreveu um sistema de Monitoramento de Atividades de Natação com RFID, utilizando o *kit* de desenvolvimento da *Nordic*, na faixa de frequência de 2.4GHz. Após testes realizados e conclusões feitas, foram levantados alguns pontos que poderiam ser implementados em propostas de trabalhos futuros:

- Desenvolver um *software* que faça programas treinos mais específicos, considerando o estilo e tipo de nado;
- Como o sistema RFID pode ler mais de uma ID de *textitag* o MAN poderia acompanhar

mais de um nadador por raia ao mesmo tempo, esta situação não foi analisada durante os testes realizados e poderia ser implementado em trabalhos futuros;

- Baseado nos treinos realizados que ficam salvos no banco de dados, podem ser criado *softwares* que gerem, de maneira gráfica, a evolução do desempenho do atleta. A exposição dos dados dessa forma ficaria mais dinâmico para o usuário.
- Uma das informações emitidas pela *tag* é a potência do sinal. Baseado nesta potência, poderia ser implementado uma rotina que faça o controle de potência, por exemplo, considerar um limite de potência aceitável e ler somente as *tags* que emitam sinais que estejam neste limite de potências;
- Desenvolver um sistema que possa fazer o nadador dar início no treino, por exemplo, um botão que fique na borda da piscina e seja pressionado pelo nadador quando o mesmo for iniciar o treino, para começar a realizar o acompanhamento;

Estes tópicos levantados são interessantes para a melhoria do sistema MAN. Caso os tópicos citados sejam implementados, o sistema pode chegar a ser um protótipo funcional.

Lista de Abreviaturas

APT *Advanced Packing Tool*

CBDA Confederação Brasileira de Desportos Aquáticos

CEFID Centro de Ciências da Saúde e do Esporte

DER Diagrama Entidade Relacionamento

FINA Federação Internacional de Natação

GPL *General Public Licence*

HF *High Frequency* - Alta Frequência

HTML *Hypertext Markup Language*

HTTP *Hypertext Transfer Protocol*

ID Identificação

IP *Internet Protocol* - Protocolo de *Internet*

JVM *Java Virtual Machine* - Máquina Virtual Java

LF *Low Frequency* - Baixa Frequência

MAN Monitoramento de Atividades de Natação

MER Modelo Entidade Relacionamento

MW *Microwave* - Microondas

PHP *Hypertext Processor*

RAM *Random Access Memory* - Memória de Acesso Aleatório

RFID *Radio-Frequency Identification*

SAW *Surface Acoustic Wave*

SGBD Sistema de Gerenciamento de Banco de Dados

SoC *System-on-chip*

SSL *Secure Sockets Layer* - Protocolo de Camada Socket Segura

UDESC Universidade do Estado de Santa Catarina

UHF *Ultra High Frequency* - Ultra Alta Frequência

XML *Extensible Markup Language*

Referências Bibliográficas

- ALBUQUERQUE, A. A. d.; MAESTRELLI, M. *Web-Server Seguro: Apache*. 2000. <http://www.rederio.br/downloads/pdf/nt00900.pdf>. Acessado em 16 de agosto de 2014.
- ALMEIDA, J. H. M. d. *PHP Com MySQL*. 2004. http://www.inf.ufsc.br/~fristtram/2464_php_com_mysql.pdf. Acessado em 15 de julho de 2014.
- BORGES, L. E. *Python Para Desenvolvedores - 2ª Edição*. [S.l.]: Edição DO Autor, 2010. ISBN 978-85-909451-1-6.
- DONIAK, M. H.; GREFF, P. d. A. Sistema de monitoramento de atividades de natação. *Pedido de Patente numero PI0904947-9A2, junto a Instituto Nacional de Propriedade Industrial*, 2009. Acessado em 16 de agosto de 2014.
- FALCAO, M. Reagras de natação 2009-2013. *Associação de Árbitros de Natação de Santa Catarina*, 2009. Acessado em 17 de agosto de 2014.
- FINKENZELLER, K. *RFID Handbook*. [S.l.]: Wiley, 2010. ISBN 978-0-470-69506-7.
- FREITAS, M. d. R. *Aperfeiçoamento em Natação: Estrutura e Organização*. [S.l.]: Syngress Publishing inc., 1999.
- GARMIN. *Forerunner 910XT - Manual do Usuario*. 2013. http://static.garmincdn.com/pumac/Forerunner_910XT_OM_PT.pdf. Acessado em 7 de julho de 2013.
- HEUSER, C. A. *Projeto de Banco de Dados - 4ª Edição*. [S.l.]: Bookman, 1998. ISBN 8577803821.
- JONES, E. C.; CHUNG, C. A. *RFID in Logistics: A Practical Introduction*. [S.l.]: CRC Press, 2008. ISBN 978-0-8493-8526-1.
- LAUERANO, M. *Programando em C para Linux, Unix e Windows*. [S.l.]: Brasport, 2005. ISBN 85-7452-233-3.
- MELO, A. A. d.; NASCIMENTO, M. G. F. *PHP Profissional*. [S.l.]: Novatec, 2007. ISBN 978-85-7522-141-9.
- MIGUENS, A. P. *Navegação Eletrônica e Em condições Espaciais - Volume III*. 2000. https://www.mar.mil.br/dhn/bhmn/publica_manualnav3.html. Acessado em 17 de agosto de 2014.
- MYSQL. *Manual de Referencia do MySQL 4.1*. 2010. <http://download.mysql.com/docs/refman-4.1-pt.a4.pdf>. Acessado em 9 de julho de 2013.

- NEVES, P.; RUAS, R. *Guia Prático do MySQL*. [S.l.]: Centro Atlântico, 2005. ISBN 972-615-006-0.
- NOLESCO, V. P.; PAVEL, R. d. C.; MOURA, R. D. Natação. *Atlas dos Esportes*, 2006. Acessado em 16 de agosto de 2014.
- PINHO, B. A. d. Avaliação do componente nrf51 da nordic visando sistemas de medições de hipismo. *Instituto Federal de Santa Catarina - Trabalho de Conclusão de Curso*, 2014. Acessado em 16 de agosto de 2014.
- RAULINO, M. F. Uso de rfid no monitoramento de atividades de natação. *Instituto Federal de Santa Catarina - Trabalho de Conclusão de Curso*, 2013. Acessado em 16 de agosto de 2014.
- RODEL, R. G. *Treinamento Resistivo Para Natação Competitiva: Revisão*. 2011. Acessado em 16 de agosto de 2014. Disponível em: <<http://hdl.handle.net/10183/39253>>.
- ROMERO, J. O tempo de reação. 2012. Acessado em 7 de julho de 2013.
- ROSA, L. A. Aplicação do rfid na cadeia logistica. *Escola Politécnica da Universidade de São Paulo - MBA em Tecnologia da Informação*, 2006.
- SALETAN, W. *Untouchable*. 2008. http://www.slate.com/articles/health_and_science/human_nature/2008/08/untouchable.html. Acessado em 16 de julho de 2013.
- SANGHERA, P. *RFID+ Study Guide and Practice Exam*. [S.l.]: Syngress Publishing inc., 2007. ISBN 978-1-59749-134-1.
- SILVA, M. S. *A Biblioteca do Programador JavaScript*. [S.l.]: Novatec, 2008. ISBN 978-85-7522-237-9.
- TAKAI, O. K.; ITALIANO, I. C.; FERREIRA, J. E. Introdução a banco de dados. *Instituto de Matemática e Engenharia - Universidade de São Paulo*, 2005. Acessado em 16 de agosto de 2014.
- VILLATE, J. E. *Física 1 - Dinâmica*. [S.l.]: Universidade do Porto, 2012. ISBN 978-972-99396-1-7.