

UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA MECÂNICA E DE COMPUTAÇÃO

DANIEL PIRES GOMES
LETÍCIA MORINAGA MATIDA

**DESENVOLVIMENTO DE UM PROTÓTIPO PARA CRONOMETRAGEM DE
PROVAS DE CORRIDA COM IDENTIFICAÇÃO POR RÁDIO FREQUÊNCIA
PARA TESTES DE APTIDÃO FÍSICA DO CENTRO DE SELEÇÃO - UFG**

GOIÂNIA

2019

**TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR
VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE
GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG**

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC nº 1204/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.


1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG):

Nome completo do autor: Daniel Pires Gomes
Letícia Morinaga Matida

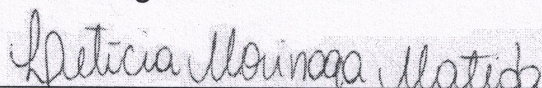
Título do trabalho: DESENVOLVIMENTO DE UM PROTÓTIPO PARA CRONOMETRAGEM DE PROVAS DE CORRIDA COM IDENTIFICAÇÃO POR RÁDIO FREQUÊNCIA PARA TESTES DE APTIDÃO FÍSICA DO CENTRO DE SELEÇÃO - UFG

2. Informações de acesso ao documento:

Concorda com a liberação total do documento SIM [] NÃO¹
Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF do TCCG.

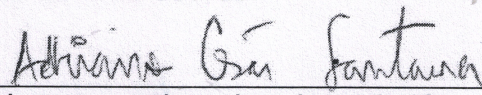


(Nome completo do autor)²



(Nome completo do autor)²

Ciente e de acordo:



(Nome completo do orientador)²

Data: 18 / 12 / 2019

¹ Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Versão abril de 2018

² As assinaturas devem ser originais sendo assinadas no próprio documento, imagens coladas não serão aceitas.

DANIEL PIRES GOMES
LETÍCIA MORINAGA MATIDA

**DESENVOLVIMENTO DE UM PROTÓTIPO PARA CRONOMETRAGEM DE
PROVAS DE CORRIDA COM IDENTIFICAÇÃO POR RÁDIO FREQUÊNCIA
PARA TESTES DE APTIDÃO FÍSICA DO CENTRO DE SELEÇÃO - UFG**

Projeto Final de Curso apresentado à Escola de Engenharia Elétrica, Mecânica e de Computação da Universidade Federal de Goiás, como requisito parcial para a integralização do curso de Engenharia Elétrica.

Orientador: Prof. Dr. Adriano César Santana

GOIÂNIA

2019

Ficha de identificação da obra elaborada pelo autor, através do Programa de
Geração Automática do Sistema de Bibliotecas da UFG.

Matida, Letícia Morinaga

DESENVOLVIMENTO DE UM PROTÓTIPO PARA
CRONOMETRAGEM DE PROVAS DE CORRIDA COM
IDENTIFICAÇÃO POR RÁDIO FREQUÊNCIA PARA TESTES DE
APTIDÃO FÍSICA DO CENTRO DE SELEÇÃO - UFG [manuscrito] /
Letícia Morinaga Matida, Daniel Pires Gomes . - 2019.

liv, 56 f.: il.

Orientador: Prof. Dr. Adraino César Santana.

Trabalho de Conclusão de Curso (Graduação) - Universidade
Federal de Goiás, , Engenharia Elétrica, Goiânia, 2019.

Bibliografia. Apêndice.

Inclui siglas, tabelas, algoritmos, lista de figuras, lista de tabelas.

1. Radio frequency identification (RFID). 2. Physical Fitness Test.
3. Timing. 4. Arduino. 5. Prototype. I. , Daniel Pires Gomes. II.
Santana, Adraino César, orient. III. Título.

CDU 621.3

DANIEL PIRES GOMES
LETÍCIA MORINAGA MATIDA

**DESENVOLVIMENTO DE UM PROTÓTIPO PARA CRONOMETRAGEM DE
PROVAS DE CORRIDA COM IDENTIFICAÇÃO POR RÁDIO FREQUÊNCIA
PARA TESTES DE APTIDÃO FÍSICA DO CENTRO DE SELEÇÃO - UFG**

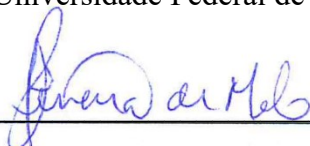
Projeto Final de Curso apresentado à Escola de Engenharia Elétrica, Mecânica e de Computação da Universidade Federal de Goiás, como requisito parcial para a integralização do curso de Engenharia Elétrica.

Goiânia, 17 de dezembro de 2019

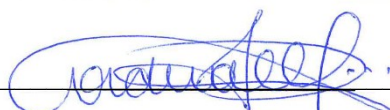
BANCA EXAMINADORA



Prof. Dr. Adriano César Santana
Universidade Federal de Goiás



Prof. Dr. Fernando Ferreira de Melo
Universidade Federal de Goiás



Prof. Dr. Leandro Luís Galdino de Oliveira
Universidade Federal de Goiás

DEDICATÓRIA

À minha família e amigos, pela dedicação e incentivo. Aos meus pais e irmãos por me encorajarem desde o início. À minha namorada Ariel Alves de Souza, por todo apoio e carinho. Ao meu parceiro e amigo Frederico Margues da Costa por todo suporte e entusiasmo.

Daniel Pires Gomes

A Deus, pela eterna graça e bondade, que permitiu a realização deste trabalho. Aos meus pais, por me apoiarem desde o início nessa jornada. À minha família e amigos, pela dedicação e carinho.

Letícia Morinaga Matida

AGRADECIMENTOS

Ao Prof. Dr. Adriano César Santana pela oportunidade, disponibilidade e dedicação, durante a realização deste Projeto Final de Curso na área de Engenharia Elétrica.

À Coordenação e aos Professores do curso de Engenharia Elétrica da Escola de Engenharia Elétrica, Mecânica e de Computação da Universidade Federal de Goiás, pelas oportunidades proporcionadas durante o curso de graduação.

E, finalmente, aos nossos familiares e amigos que sempre estiveram ao nosso lado nos apoiando e incentivando em todos os momentos dessa jornada e a todos àqueles que, direta ou indiretamente, tenham contribuído para o desenvolvimento deste trabalho.

RESUMO

Neste trabalho é ilustrado uma breve aplicação da tecnologia de identificação por rádio frequência (RFID), utilizada para o desenvolvimento de um sistema de cronometragem de provas de corrida. Inicialmente é retratada uma contextualização seguida de uma explicação do seu funcionamento e dos componentes necessários para completar o sistema. O sistema automatizado de cronometragem é uma necessidade do Centro de Seleção da Universidade Federal de Goiás (CS - UFG), que é responsável por aplicar diversas provas de concursos públicos, entre elas os Testes de Aptidão Física (TAF), que são avaliações exigidas para alguns cargos públicos, como policiais e bombeiros. A solução proposta é o desenvolvimento de um protótipo de sistema automatizado para cronometragem de provas de corridas, capaz de aumentar a precisão na marcação de tempo, facilitar o gerenciamento de candidatos, aquisição de dados e divulgação dos relatórios de resultados. Ou seja, garantir maior segurança, reduzir custos, probabilidade de erros por fator humano e agilizar a aplicação desta etapa do TAF nos concursos realizados pelo CS-UFG. Para o desenvolvimento deste sistema foram utilizadas placas Arduino UNO, módulos MFRC522 e RTC DS3231, protocolos de comunicação, aplicativo Excel do pacote Office e as plataformas de desenvolvimento IDE Arduino e IDE Visual Studio. O método Projeto Iterativo foi empregado para o desenvolvimento do protótipo, o que baseou a evolução das fases em níveis de aprimoramento. O sistema é composto por três etapas: aquisição de dados; registro de dados e tratamento de dados. Os resultados obtidos foram: um conjunto para aquisição de dados; um software denominado CRONÔMETRO RFID com interface para o usuário, funcionalidades de configuração de dispositivos e registro dos dados; e uma planilha para o tratamento de dados e geração de relatórios de resultados. As etapas de registros de dados e tratamentos de dados foram avaliadas como protótipos nível beta, as quais são hábeis para aplicação de testes reais. Porém, a etapa de aquisição de dados configurou-se como protótipo alfa, pois o alcance de leitura obtido não foi satisfatório. No entanto, comprova a aplicabilidade da tecnologia RFID para cronometragem de corridas.

Palavras Chave:

Cronometragem; Identificação por rádio frequência (RFID); Arduino; Protótipo; Corrida; Teste de Aptidão Física (TAF).

ABSTRACT

In this paper a brief application of radio frequency identification technology (RFID), used for the development of a race test timing system, is illustrated. Initially a contextualization, followed by an explanation of its operation and the components needed to complete the system. The automated timing system is a necessity of the Selection Center of the Federal University of Goiás (CS - UFG), which is responsible for applying several tests of public tenders, among them the Physical Fitness Tests (TAF), which are evaluations required for some public office such as police and fire brigade. The proposed solution is the development of an automated race test timing system prototype that can increase accuracy in timekeeping, facilitate candidate management, data acquisition and dissemination of results reports. That is, ensure greater security, reduce costs, probability of errors by human factor and expedite the application of this stage of TAF in the contests held by CS-UFG. Arduino UNO cards, MFRC522 and RTC DS3231 modules, communication protocols, Office suite Excel application, and Arduino IDE and Visual Studio IDE development platforms were used to develop this system. The Iterative Design method was used for prototype development, which based the evolution of the phases on levels of improvement. The system consists of three steps: data acquisition; data recording and data processing. The results obtained were: a set for data acquisition; software called “CRONÔMETRO RFID” with user interface, device configuration and data logging functionality; and a spreadsheet for data processing and reporting of results. The data recording and data processing steps were evaluated as beta level prototypes, which are able to apply in real tests. However, the data acquisition stage was stipulated as an alpha prototype, because the reading range obtained was not satisfactory. However, it proves the applicability of RFID technology for race timing.

Key Words:

Timing; Radio frequency identification (RFID); Arduino; Prototype; Running; Physical Fitness Test.

LISTA DE FIGURAS

Figura 1 – Linha do tempo RFID.....	15
Figura 2 – PINOUT Arduino UNO R3.....	17
Figura 3 – Esquema de ligação Mestre-Escravo.....	19
Figura 4 – PINOUT MFRC522.....	20
Figura 5 – Cartão e Chaveiro S50 MIFARE.....	21
Figura 6 – PINOUT RTC DS3231.....	22
Figura 7 – Arduino IDE.....	23
Figura 8 – Microsoft Visual Studio.....	24
Figura 9 – Projeto aberto na IDE.....	25
Figura 10 – Diagrama relação de compilação e execução em C#.....	26
Figura 11 – Microsoft Excel.....	27
Figura 12 – Diagrama geral.....	28
Figura 13 – Dispositivos físicos.....	29
Figura 14 – Arquitetura de comunicação.....	30
Figura 15 – Conjunto placa Arduino UNO, RFID-RC522 e RTC DS3231.....	30
Figura 16 – Fluxograma código aquisição de dados.....	31
Figura 17 – Tela inicial CRONÔMETRO RFID.....	33
Figura 18 – Edição código CRONÔMETRO RFID.....	33
Figura 19 – Edição interface gráfica CRONÔMETRO RFID.....	34
Figura 20 – Fluxograma código fonte software CRONÔMETRO RFID.....	34
Figura 21 – Criar eventos.....	35
Figura 22 – Dispositivos.....	36
Figura 23 - Diretório.....	36
Figura 24 – Mensagem diretório vazio.....	37
Figura 25 – Sincronizar tempo.....	38
Figura 26 – Modo Registro.....	39

Figura 27 – SALVAR LOG DE REGISTROS.....	39
Figura 28 – Modo Corrida.....	40
Figura 29 – SALVAR LOG DE CORRIDA.....	41
Figura 30 – Fluxograma código fonte planilha de tratamento de dados.....	42
Figura 31 – Aba de interface com o usuário.....	43
Figura 32 – Banco de dados após carregar dados de registros.....	43
Figura 33 – Banco de dados após carregar dados de corrida.....	44
Figura 34 – Campos de filtros.....	44
Figura 35 – Tempo de prova e botão gerar lista de resultado.....	45
Figura 36 – Tempo de prova e botão gerar lista de resultado.....	45
Figura 37 – Página padrão para exportar ou imprimir lista de resultado.....	45
Figura 38 – Conjunto placa Arduino UNO, RFID-RC522 e RTC DS3231.....	47
Figura 39 – Estabelecimento de comunicação serial.....	47
Figura 40 – Registro de candidatos para prova de corrida e de aquisição de dados.....	48
Figura 41 – arquivos exportados e registros.....	48
Figura 42 – arquivos exportados e registros.....	49
Figura 43 – arquivos exportados e registros.....	49
Figura 44 – Alterações ganho de sinal módulo MFRC 522.....	50
Figura 45 – Alterações ganho de sinal módulo MFRC 522.....	50
Figura 46 – Calculadora de parâmetros para bobina de várias camadas com núcleo de ar.....	51
Figura 47 – Calculadora de bobina com núcleo de ar de várias camadas.....	52

LISTA DE SIGLAS

AC/DC	<i>Alternating Current/Direct Current</i>
AM/PM	<i>Ante Meridiem/Post Meridiem</i>
BD	Banco de dados
CLI	<i>Common Language Infrastructure</i>
CLR	<i>Common Language Runtime</i>
CS - UFG	Centro de Seleção da Universidade Federal de Goiás
DLL	<i>Dynamic-link library</i>
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
EXE	<i>Executable</i>
FDX	<i>Full-Duplex</i>
HDX	<i>Half-Duplex</i>
I2C	<i>Inter Integrated Circuit</i>
IDE	<i>Integrated Development Environment</i>
IFF	<i>Identifier Friend-Foe</i>
IL	<i>Intermediate Language</i>
IoT	<i>Internet of Things</i>
JIT	<i>Just in Time</i>
MISO	<i>Master In, Slave Out</i>
MOSI	<i>Master Out, Slave In</i>
NFC	<i>Near Field Communication</i>
PDF	<i>Portable Document Format</i>
RFID	<i>Radio Frequency Identification</i>
RTC	<i>Real Time Clock</i>
SCLK	<i>Serial Clock</i>
SEQ	<i>Sequential</i>
SPI	<i>Serial Peripheral Interface</i>
SS	<i>Slave Select</i>
TAF	Teste de Aptidão Física
UART	<i>Universal Asynchronous Receiver Transmitter</i>
UID	<i>Unique Identifier</i>
USB	<i>Universal Serial Bus</i>
VBA	<i>Visual Basic Application</i>

SUMÁRIO

1 - INTRODUÇÃO.....	14
1.1 Justificativa.....	16
1.2 Objetivo.....	16
2 – MATERIAIS E MÉTODOS.....	17
2.1 Métodos.....	17
2.2 Materiais.....	17
2.2.1 Arduino UNO.....	17
2.2.2 Protocolo de Comunicação Serial.....	18
2.2.3 Módulo RFID MFRC522.....	19
2.2.4 Comunicação RFID.....	20
2.2.5 Cartão/ tag RFID MIFARE.....	21
2.2.6 Módulo RTC DS3231.....	22
2.3 Plataformas de desenvolvimento e linguagens de programação.....	23
2.3.1 Arduino IDE e C++.....	23
2.3.2 IDE Visual Studio e C#.....	24
2.3.3 Microsoft Office Excel e VBA.....	26
3 - DESENVOLVIMENTO.....	28
3.1 Configurações gerais.....	28
3.2 Configurações físicas.....	29
3.3 Arquitetura de comunicação.....	29
3.4 Aquisição de dados.....	30
3.5 Registro de dados.....	32
3.5.1 Criar eventos.....	35
3.5.2 Dispositivos.....	35
3.5.3 Diretório.....	36
3.5.4 Sincronizar Tempo.....	37

3.5.5 Modo Registro.....	38
3.5.6 Modo Corrida.....	40
3.6 Tratamento de dados no Excel.....	41
3.6.1 Importar dados.....	43
3.6.2 Listas de resultados.....	44
4 – RESULTADOS E DISCUSSÃO.....	46
5 - CONCLUSÃO.....	53
REFERÊNCIAS.....	54
APÊNDICES.....	56
APÊNDICE A – Código utilizados.....	56

1 - INTRODUÇÃO

Durante a 3ª Revolução Industrial, em meados da década de 60, com a evolução da microeletrônica surgiram os microcontroladores, dispositivos capazes de realizar transmissão e tratamento de dados envolvendo lógicas digitais e/ou analógicas, com capacidade de armazenamento. O surgimento desses dispositivos revolucionou não apenas a automação da indústria, mas trouxe a tecnologia também para os setores de bens e serviço (MORAES & CASTRUCCI, 2007; SILEVIRA & LIMA, 2003).

O desenvolvimento dos microcontroladores remontam um longo processo que vai desde as criações dos transistores IBM 7090 em 1960 até os mais atuais como Raspberry pi (ARM) em 2011. Durante esse processo muitos microcontroladores foram criados, o primeiro foi o INTEL 4004 da fabricante Intel que surgiu em 1971 com memória RAM de 4 bits. A Intel, em seguida lançou os microprocessadores e microcontroladores da linha MCS, dos quais o MCS-51 ou 8051, com memória RAM de 8 bits, lançado em 1980, foi o maior influenciador no cenário de mercado (ZILLER, 2000).

Outro marco importante na história dos microcontroladores foi a chegada ao mercado dos PIC's, lançado em 1985 pela fabricante *General Instruments*, em sua primeira versão de 8 bits, atualmente existem opções de memória de 16 e 32 bits. Entre o surgimento do 8051 e do PIC, entrou no mercado o ARM da *ARM Holdings*, primeiro microcontrolador de 32 bits que evoluiu para a placa Raspberry pi, e em 1996 surgiu o Atmel AVR de 8 bits que evoluiu para as placas Arduino comercializadas atualmente (ZILLER, 2000).

Os microcontroladores se tornaram imprescindíveis no mundo moderno e são os maiores responsáveis pela movimentação da indústria de semicondutores no país. Essa importância se dá pelas inúmeras possibilidades de aplicação na indústria, esporte, meio ambiente, segurança, saúde entre outras. Além disso geram grande impacto na cultura, educação e principalmente na dinamicidade do setor de tecnologias (CHANDLER, 2002).

Considerando que o mundo caminha para a 4ª Revolução Industrial nota-se o empenho de grandes empresas de produção de tecnologia como a Microsoft, Arduino e Apple para disseminar o conhecimento e as ferramentas necessárias para que a maioria da população tenha acesso a chamada tecnologia 4.0 (ORTEGA, 2019).

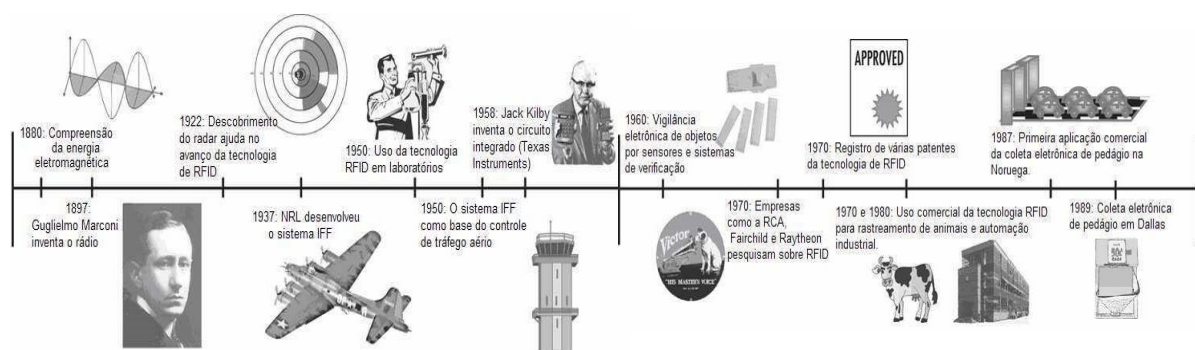
Uma exemplificação disso é toda linha de softwares e hardwares da Arduino que por anos possuem a maior aplicabilidade de seus produtos em projetos por todo o mundo. A empresa surgiu no *Ivrea Interaction Design Institute* para ser uma ferramenta simples de prototipagem. Com a expansão de seu público alvo e adaptando-se às novas tecnologias

iniciou a oferta de produtos voltados a *Internet of Things* (IoT), *wearable* e impressão 3D. As placas Arduino possuem todos os seus códigos divulgados e acessíveis para desenvolvedores independentes que queiram fabricar suas próprias placas, modificá-las ou implementar funcionalidades àquelas já existentes. O software possui código aberto, dessa forma usuários de todo o mundo contribuem para aumentar o acervo de bibliotecas e funcionalidades (ARDUINO, 2019).

O conceito prático de globalização e o sistema capitalista também influenciam no surgimento de novas tecnologias, implicando na sociedade as ideias de agilidade e melhor aproveitamento de tempo. Em meio a esse cenário as tecnologias *wireless* conquistaram seu espaço e continuam expandindo suas aplicações, um exemplo é a ascensão das tecnologias *Near Field Communication* (NFC) e *Radio Frequency Identification* (RFID) que possuem aplicações nas mais diversas áreas da sociedade (GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO, 2018).

A tecnologia RFID surgiu de uma pesquisa inglesa que produziu o *Identifier Friend or Foe* (IFF) que foi utilizada durante a Segunda Guerra para identificar aviões inimigos. A partir dessa tecnologia foram desenvolvidas diversas pesquisas que resultaram nos dispositivos de vigilância e identificação de objetos em 1960, primeiro protótipo com tecnologia RFID. Desde então surgiram novas aplicações e avanços nessa tecnologia que permitiram sua ascensão na atualidade, esse histórico de desenvolvimento está demonstrado na Figura 1 (GONSALES, 2017).

Figura 1 – Linha do tempo RFID



Fonte: GONSALES, 2017.

A tecnologia RFID está presente em várias áreas da sociedade, como sua utilização em controle de estoques e rastreamento de mercadorias, tarifamento de pedágios e estacionamento, controle de localização e identificação de gado em grandes fazendas. Um exemplo de aplicação importante do RFID presente no Brasil é o sistema de cronometragem presente em diversos circuitos de corridas importantes no país (GONSALES, 2017).

1.1 Justificativa

De acordo com o escopo no artigo 37, inciso II da Constituição Federal:

II - A investidura em cargo ou emprego público depende de aprovação prévia em concurso público de provas ou de provas e títulos, de acordo com a natureza e a complexidade do cargo ou emprego, na forma prevista em lei, ressalvadas as nomeações para cargo em comissão declarado em lei de livre nomeação e exoneração.

A ocupação de cargos públicos depende de aprovação em concurso público, que são compostos por provas de avaliação de acordo com a natureza do cargo. Uma das etapas de avaliação para cargos policiais, bombeiros e outros são os Testes de Aptidão Física (TAF). Um dos testes que normalmente compõe um TAF é a corrida, onde o candidato deve ser capaz de percorrer uma distância definida em um intervalo de tempo estipulado.

O Centro de Seleção da Universidade Federal de Goiás (CS - UFG) é responsável por aplicar diversas provas de concursos públicos, incluindo a etapa de TAF. No entanto, o atual método utilizado para a cronometragem das provas de corrida é o manual, ou seja, os candidatos são separados em grupos de três e um fiscal de prova é responsável por marcar a distância percorrida e o tempo de prova. Processo este que devido ao grande número de candidatos a serem avaliados requer numerosa quantidade de fiscais de prova. Por isso, este método é dispendioso e suscetível a erros por fator humano.

1.2 Objetivo

O objetivo do trabalho é desenvolver um protótipo de sistema automatizado para cronometragem de provas de corridas em Testes de Aptidão Física aplicados pelo CS - UFG. O sistema deve ser capaz de aumentar a precisão na cronometragem, facilitar o gerenciamento de candidatos, aquisição de dados e divulgação dos relatórios de resultados. Ou seja, garantir maior segurança, reduzir custos, probabilidade de erros por fator humano e agilizar a aplicação desta etapa do TAF nos concursos realizados pelo CS-UFG.

2 – MATERIAIS E MÉTODOS

2.1 Métodos

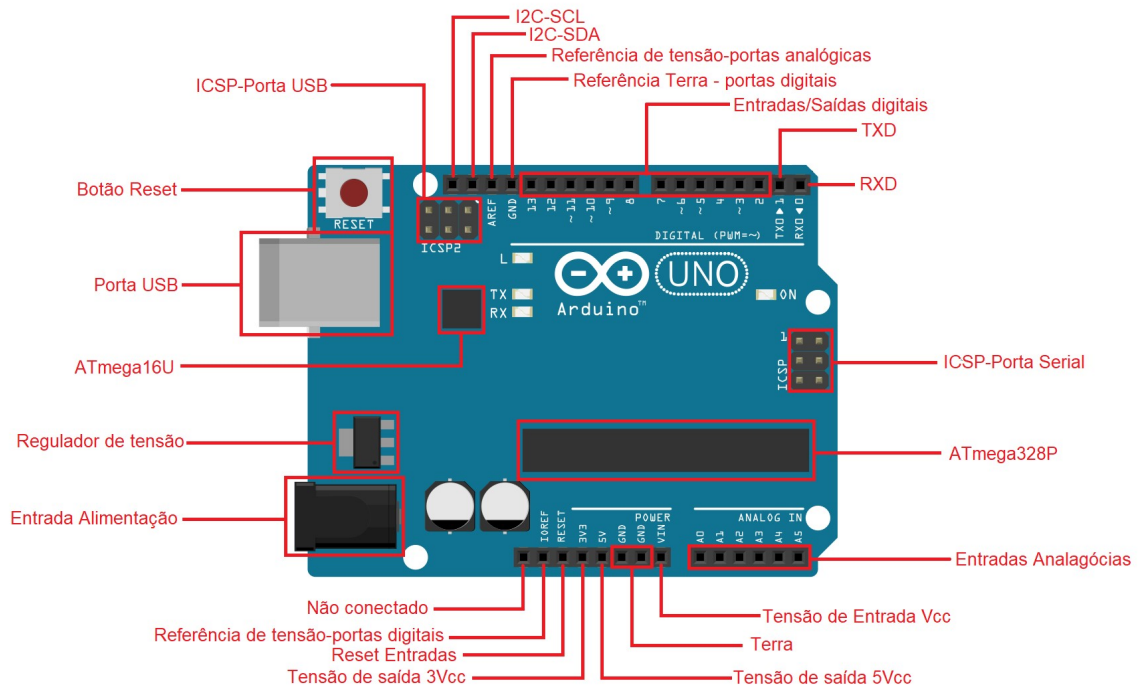
Neste projeto foi aplicada a seguinte metodologia: Levantamento bibliográfico e fundamentação teórica; Estudo dos protocolos de comunicação; Estudo dos manuais técnicos dos módulos e softwares dos fabricantes; Definição da arquitetura de comunicação a ser implementada; Programação dos dispositivos; Criação do Software; Implementação e Testes.

2.2 Materiais

2.2.1 Arduino UNO

O Arduino UNO R3 é uma placa de microcontrolador baseado no ATmega328P, possui ampla aplicação nos campos didáticos e de desenvolvimento livre, por se tratar de um *hardware* de código aberto. A placa possui 14 entradas/saídas digitais, 6 entradas analógicas e um cristal de 16MHz, é implementada na placa uma entrada para comunicação serial via cabo USB (*Universal Serial Bus*) e um adaptador AC/DC (*Alternating Current/Direct Current*) para tomada, conforme mostrado na Figura 2 (ARDUINO, 2019).

Figura 2 – PINOUT Arduino UNO R3



Fonte: AUTORIA PRÓPRIA, 2019.

A programação do Arduino UNO R3 é realizada no software Arduino IDE (*Integrated Development Environment*), que é um ambiente de desenvolvimento e programação integrado com suporte para todas as placas Arduino e possui suporte para Windows, Linux, e MAC OS X. O ATmega 328P possui suporte de comunicação serial via protocolos SPI, UART e I2C (*Serial Peripheral Interface, Universal Asynchronous Receiver Transmitter, Inter Integrated Circuit*, respectivamente), de forma que é possível utilizar módulos adicionais na placa que suportem a mesma comunicação. Por ser uma placa de código aberto existe uma vasta quantidade de códigos, bibliotecas, e exemplos na Internet para auxílio na programação dela, além de possuir manuais de programação no site da Arduino (ARDUINO, 2019).

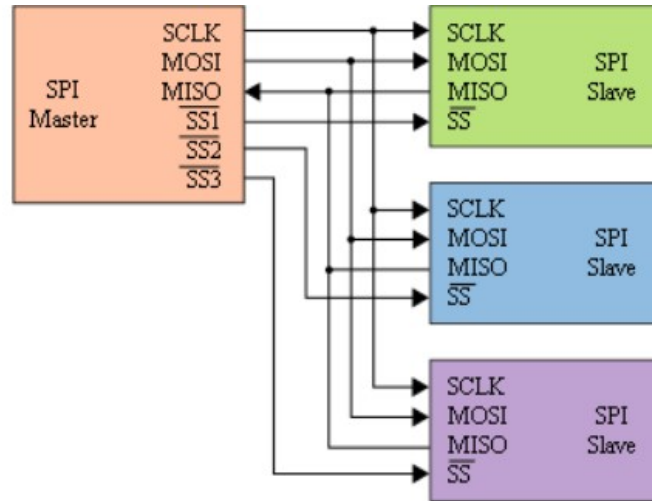
2.2.2 Protocolo de Comunicação Serial

Comunicação Serial é o processo onde os *bits* são enviados individualmente em série por um barramento ou canal de comunicação podendo ser do tipo síncrona ou assíncrona. Na assíncrona os *bits* são enviados continuamente pelo canal de comunicação ou barramento sem intervalos possibilitando a sincronização entre transmissor e receptor. Com a síncrona, a transmissão ocorre caractere por caractere, dessa forma, o canal de comunicação ou barramento permanece em repouso até que seja enviado informação para o mesmo. Dentre os protocolos de comunicação serial se destacam 3 protocolos: UART, SPI e I2C (MARTINO, 2019).

A comunicação UART é do tipo assíncrona e é base de dois padrões amplamente utilizados RS232 e RS422. As portas USB utilizam o padrão de comunicação RS232 e unificam a comunicação entre dispositivos que usam esse protocolo por ser um porta padrão. Os pinos de comunicação característicos desse padrão são a TX e RX que se destinam para transmissão e recepção de dados respectivamente.

A comunicação SPI é do tipo síncrona e permite uma integração de mais de 2 dispositivos ao mesmo tempo pois opera na estrutura mestre-escravo e fluxo de dados simultâneo, conforme exemplificada na Figura 3. Essa estrutura é composta por um dispositivo central denominado como mestre e dispositivos periféricos denominados escravos que recebem “ordens” e retornam ao mestre as informações requisitadas (MENDONÇA, 2016).

Figura 3 – Esquema de ligação Mestre-Escravo



Fonte: MENDONÇA, 2016.

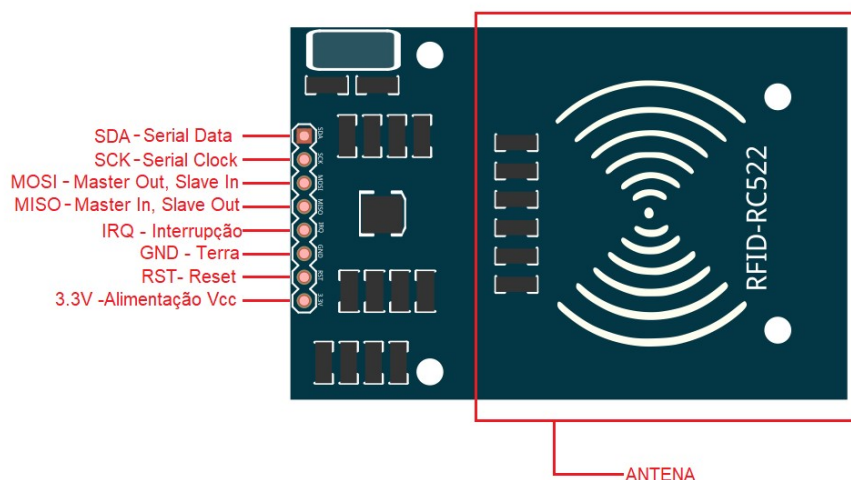
Os pinos característicos dessa comunicação são o MISO (*Master In, Slave Out*), MOSI (*Master Out, Slave In*) e SS (*Slave Select*) e são destinadas para envio de dados do escravo para o mestre, do mestre para o escravo e seleção do escravo respectivamente. O pino SCLK (*Serial clock*) é destinado a sincronização da comunicação serial entre os dispositivos.

O protocolo I2C é do tipo síncrono e funciona também com a estrutura mestre escravo, entretanto não possui uma função de seleção de escravo e sim um endereçamento para cada um direcionando as informações. Isso ocorre devido ao fato de ser uma comunicação por 2 pinos, SDA (*Serial Data*) e SCL (*Serial clock*) que se destinam ao tráfego de dados e sincronização da comunicação respectivamente. Dessa forma esse barramento ou canal de comunicação não funciona com envio e recepção das informações de forma simultânea (MENDONÇA, 2016).

2.2.3 Módulo RFID MFRC522

O módulo RFID (*Radio Frequency Identification*) MFRC522, é um módulo passivo de leitura e escrita RFID, da linha MIFARE, da empresa NXP Semiconductors. O MFRC522 possui 8 pinos, sendo 2 deles de alimentação e 6 de comunicação e possui antena integrada, conforme mostrado na Figura 4 (NX SEMICONDUCTORS, 2016).

Figura 4 – PINOUT MFRC522



Fonte: AUTORIA PRÓPRIA, 2019.

De acordo com o *Datasheet* desse equipamento, a frequência de operação da antena é 13,56 MHz, sendo compatível com *tags* que operam na mesma frequência e possuem os protocolos de escrita e estruturas de memória compatíveis com os da MIFARE. O módulo é composto por um transceptor com decodificador e uma antena que emite um sinal de rádio, ou seja, gera um campo eletromagnético. O chip decodifica o sinal recebido e o transmite para a placa Arduino UNO. O MFRC522 possui suporte de comunicação via protocolos SPI, I2C e UART, dessa forma é compatível com microcontroladores que suportam esses protocolos, como as placas Arduino e Raspberry.

2.2.4 Comunicação RFID

O RFID é uma tecnologia de aquisição e armazenamento de dados por rádio frequência. Um sistema RFID é composto por um leitor RFID, antena e *tags* /etiquetas e pode operar nas mais diversas frequências. O leitor RFID é composto por um chip, *transponder* e *decode*, e uma antena que pode ser ou não integrada ao módulo (GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO UFRJ, 2018).

As *tags* RFID podem ser do tipo passiva ou ativa. A passiva não emite um sinal independente, ela precisa estar na presença de um leitor RFID para ter seu circuito alimentado, já a ativa emite um sinal independente da presença do leitor, pois possui uma fonte de alimentação interna, uma bateria. Além disso a comunicação das *tags* pode ser dividida em: HDX (*Half-Duplex*), FDX (*Full-Duplex*) ou SEQ (*Sequential*). Na comunicação HDX é possível o envio e recebimento de dados pela *tag*, mas não de forma simultânea, já na FDX é

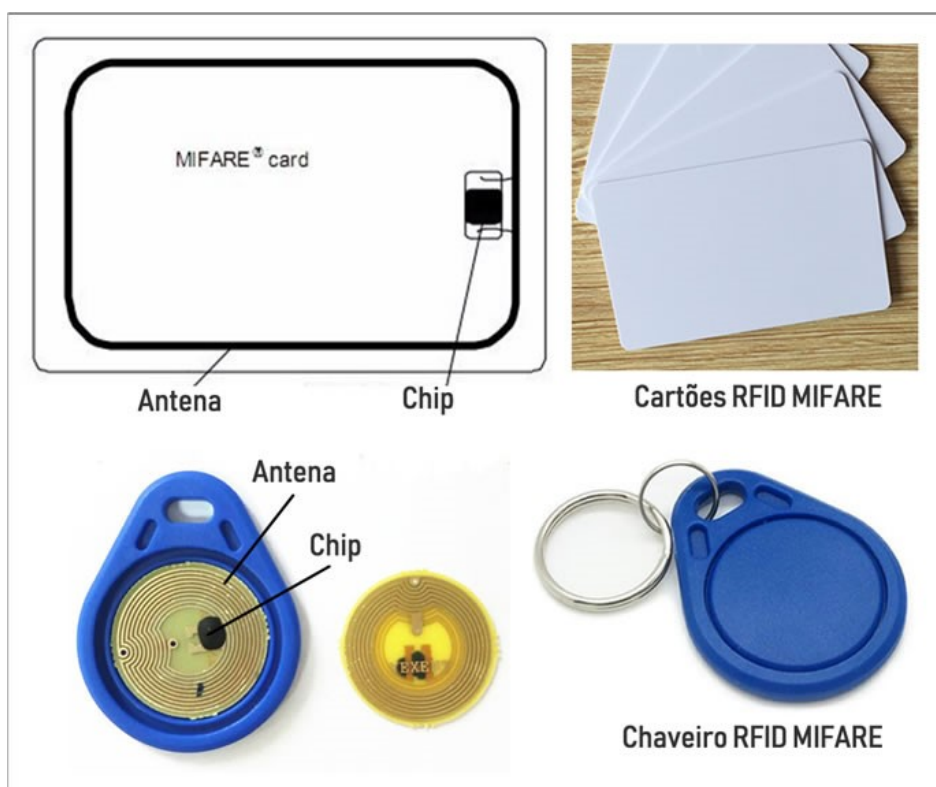
possível realizar essas ações simultaneamente. A comunicação SEQ se assemelha a FDX, porém a maior diferença está na emissão de sinal de energia para a antena/leitor que para a comunicação SEQ é realizada apenas quando há transmissão de dados (GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO, 2012).

Quanto ao tipo de memória das *tags* elas podem ser memórias de leitura onde os dados são programados uma única vez e podem ser lidos pelos usuários, leitura e escrita única na qual é possível a mudança de dados uma vez após a programação e a memória de leitura e escrita, onde pode-se gravar e escrever na memória da *tag* quantas vezes necessário (GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO, 2012).

2.2.5 Cartão/ tag RFID MIFARE

A NXP Semiconductors possui 2 tipos de *tags* RFID para o módulo MFRC-522 da linha S50: cartão e chaveiro. O cartão possui dimensões de 85 x 54 x 0,9mm, memória EEPROM (*Electrically-Erasable Programmable Read-Only Memory*) de 1Kbyte e suporte de escrita e leitura, já o chaveiro possui dimensões de 40 x 32 x 4mm e as mesmas características do cartão. As *tags* são compostas por um chip RFID e uma antena, as características físicas podem ser vistas na Figura 5.

Figura 5 – Cartão e Chaveiro S50 MIFARE



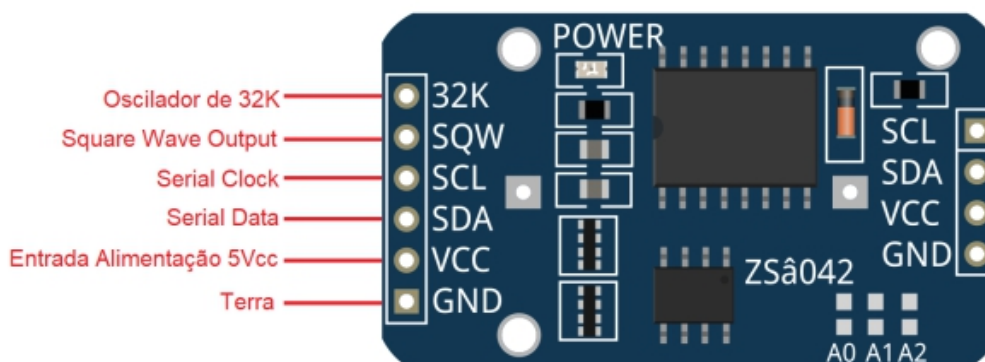
A memória EEPROM das *tags* de 1 K byte são formadas por 16 setores de 4 blocos cada e possuem estrutura de memória dividida em “*Manufacturer Block*”, “*Data Block*” e “*Sector Trailer*”. O “*Manufacturer Block*” destina-se as informações da fabricante, como versão e identificação do cartão UID (*Unique Identifier*). O “*Data Block*” é o bloco de armazenamento de dados e pode ser do tipo leitura/escrita ou valor, sendo esse último destinado a uso com contagens pois possui funções de incremento, decremento, restauração e transferência. O “*Sector Trailer*” é destinado as chaves de autenticação de escrita e acesso. O primeiro bloco do primeiro setor, bloco 0 do setor 0, é o “*Manufacturer Block*” e o último de cada setor, bloco 3, é destinado ao “*Sector Trailer*”, este tem o papel de armazenar as chaves de autenticação dos blocos de cada setor, o restante dos blocos são os blocos de dados (Mundo Eletrônica, 2019).

A leitura de dados da *tag* é realizada ao entrar em contato com o campo eletromagnético produzido pelo MRFC522, quando isso ocorre, corrente elétrica é induzida no circuito da antena da *tag*. O chip quando energizado emite um sinal contendo as informações de UID e, dependendo da *tag* utilizada, outros dados que também podem ser gravados em sua memória (MUNDO ELETRÔNICA, 2019).

2.2.6 Módulo RTC DS3231

O RTC DS3231 (*Real Time Clock DS3231*, Clock de Tempo Real) é um módulo de relógio real capaz de enviar e receber dados via protocolo I2C. O RTC DS3231 possui 10 pinos, sendo 4 deles de alimentação e 6 de comunicação conforme mostrado na Figura 6 (MAXIM INTEGRATED PRODUCTS, 2015).

Figura 6 – PINOUT RTC DS3231



Fonte: AUTORIA PRÓPRIA, 2019.

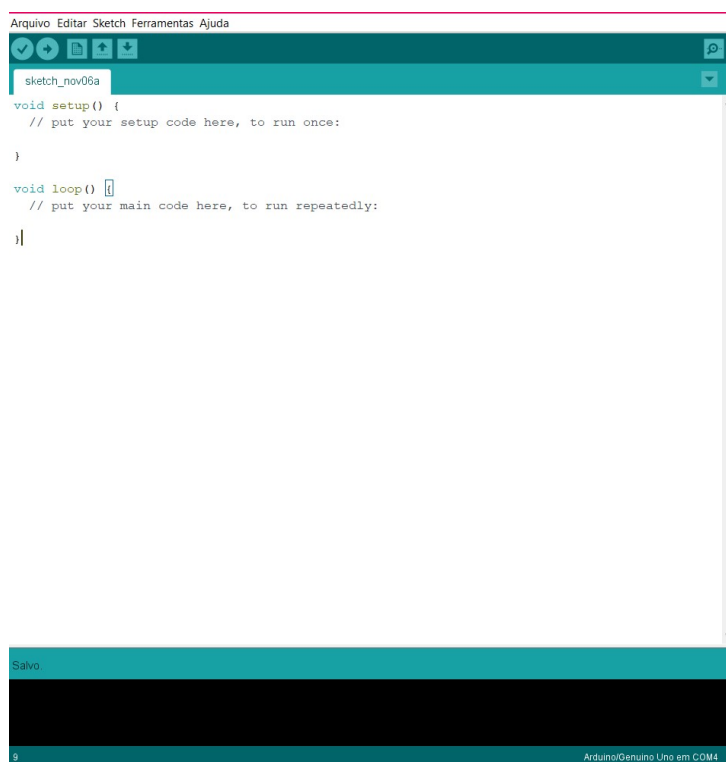
Conforme *datasheet* o DS3231 mantém informações de segundos, minutos, horas, dia, data, mês e ano, ajusta automaticamente sua configuração para anos bissextos. O módulo pode operar no formato 12 ou 24 horas com indicador AM/PM (*Ante Meridiem/Post Meridiem*) e possui 2 alarmes programáveis e suporte para comunicação I2C através de um barramento bidirecional. Além disso o módulo DS3231 possui uma entrada de bateria 3.3V de forma a suprir a alimentação do microcontrolador e manter a contagem de data e tempo caso a alimentação principal seja interrompida (MAXIM INTEGRATED PRODUCTS, 2015).

2.3 Plataformas de desenvolvimento e linguagens de programação

2.3.1 Arduino IDE e C++

O Arduino IDE (*Integrated Development Environment*), Figura 7, é um ambiente de desenvolvimento e programação integrado da Arduino com suporte para as placas e módulos desenvolvidos pela fabricante e desenvolvedores independentes e possui linguagem de programação básica baseada na linguagem C++. O software IDE Arduino é disponível para download gratuito no site da companhia, ou para uso on-line (ARDUINO, 2019).

Figura 7 – Arduino IDE



Fonte: AUTORIA PRÓPRIA, 2019.

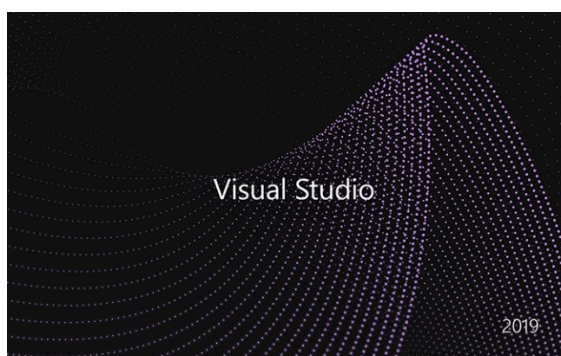
Os conteúdos fabricados pela Arduino são de código aberto, isso permite que qualquer desenvolvedor independente reproduza as placas e módulos da Arduino, implemente bibliotecas para o software desde que disponibilizado de acordo com os Termos de Uso da Marca. Dessa forma existe grande extensão de bibliotecas disponível na internet e é possível implementá-las no Arduino IDE (ARDUINO, 2019).

A linguagem de programação utilizada nesse software é baseada em C++, e possui um manual/guia de programação suportado por cada placa Arduino disponível em seu site, a programação do Arduino, que pode ser dividida em funções, valores e estruturas. As funções são as responsáveis pelo controle da placa e realização de cálculos, os valores são as variáveis que podem ser dados ou constantes, já as estruturas são os elementos do código base C++, que inclui os operadores e as estruturas de definição e estruturamento do código (ARDUINO, 2019).

2.3.2 IDE Visual Studio e C#

O ambiente de desenvolvimento utilizado para criar o software para registro de dados foi a IDE Visual Studio, Figura 8, versão *Community* 2019, disponível gratuitamente pela Microsoft para desenvolvedores de softwares em geral.

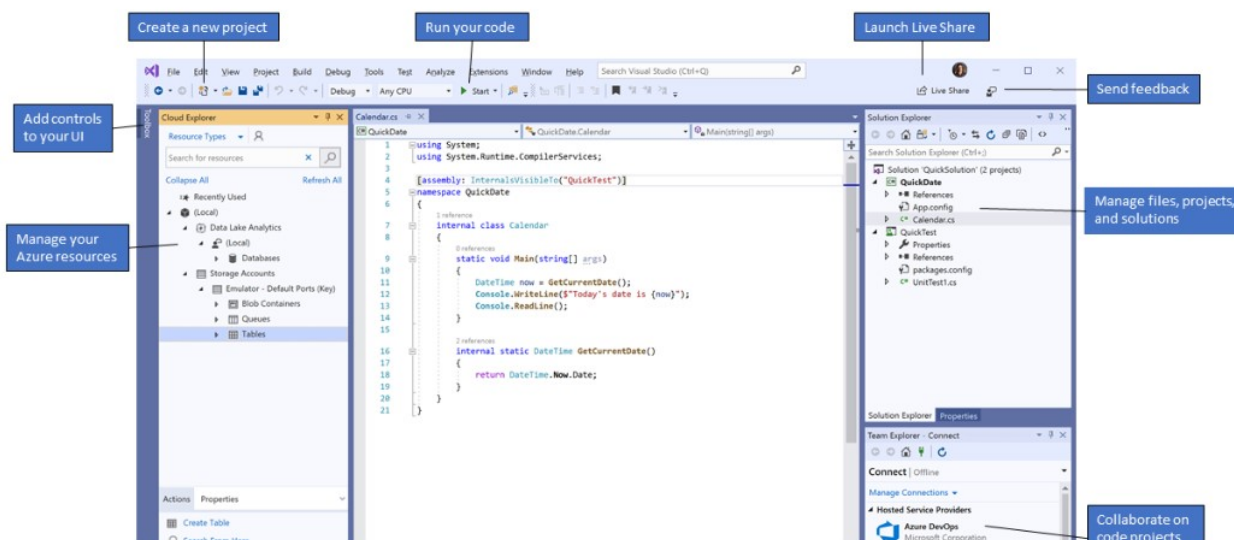
Figura 8 – Microsoft Visual Studio



Fonte: AUTORIA PRÓPRIA, 2019.

O ambiente de desenvolvimento integrado (IDE) do Visual Studio, Figura 9, é um painel de inicialização criativo que permite editar, depurar e compilar o código e, em seguida, publicar um aplicativo. Uma IDE é um programa repleto de recursos que pode ser usado por muitos aspectos no desenvolvimento de softwares. Além do editor e do depurador padrão, o Visual Studio inclui compiladores, ferramentas de preenchimento de código, designers gráficos e muitos outros recursos para facilitar o processo de desenvolvimento de software (MICROSOFT, 2019).

Figura 9 – Projeto aberto na IDE



Fonte: AUTORIA PRÓPRIA, 2019.

Esta imagem mostra o Visual Studio com um projeto aberto e várias janelas de ferramentas, como o gerenciador de soluções (parte superior direita) que permite exibir, navegar e gerenciar os arquivos de código e a janela do editor (parte central), onde exibe o conteúdo do arquivo para editar o código ou criar uma interface do usuário, como uma janela com botões e caixas de texto. A IDE permite desenvolver projetos em C#, F#, Visual Basic, C++, Python e JavaScript (MICROSOFT, 2019).

A linguagem de programação C# foi a escolhida para o desenvolvimento deste trabalho. C# é uma linguagem orientada a objeto e fortemente tipada, que permite que os desenvolvedores criem uma variedade de aplicativos robustos e seguros executados no .NET Framework. Com ela é possível criar aplicativos de cliente do Windows, serviços Web XML, componentes distribuídos, aplicativos cliente-servidor, aplicativos de banco de dados e outros (MICROSOFT, 2015).

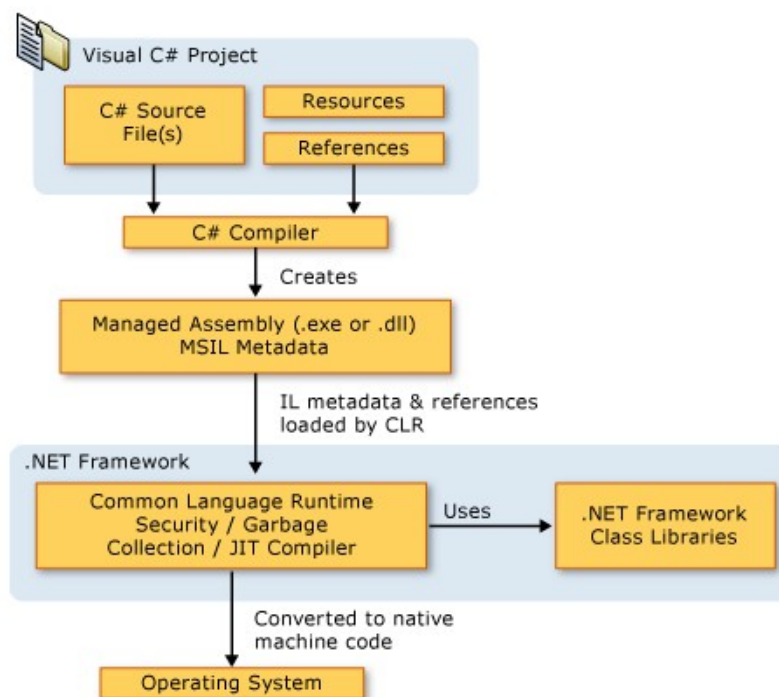
Programas em C# são executados em um componente integral do Windows, o .NET Framework, que inclui um sistema de execução virtual chamado de CLR (*Common Language Runtime*) e um conjunto unificado de bibliotecas de classes. O CLR é a implementação comercial da Microsoft da CLI (*Common Language Infrastructure*), um padrão internacional que é a base para a criação de ambientes de execução e de desenvolvimento nos quais linguagens e bibliotecas funcionam de forma integrada (MICROSOFT, 2015).

O código-fonte escrito em C# é compilado em uma IL (*Intermediate Language*) que está em conformidade com a especificação da CLI. O código e os recursos de IL, como

bitmaps e cadeias de caracteres, são armazenados em disco em um arquivo executável chamado de *assembly*, normalmente com uma extensão EXE (*executable*) ou DLL (*Dynamic-link library*). Um *assembly* contém um manifesto que fornece informações sobre os tipos, a versão, a cultura e os requisitos de segurança do *assembly* (MICROSOFT, 2015).

Quando o programa em C# é executado, o *assembly* é carregado no CLR, que pode executar várias ações de acordo com as informações no manifesto. Em seguida, se os requisitos de segurança forem atendidos, o CLR executará a compilação JIT (*just in time*) para converter o código de IL (Linguagem Intermediária) em instruções nativas da máquina. O CLR também oferece outros serviços relacionados à coleta automática de lixo, tratamento de exceções e gerenciamento de recursos. O código que é executado pelo CLR é, às vezes, chamado de "código gerenciado", ao contrário de "código não gerenciado", que é compilado em linguagem de máquina nativa e visa um sistema específico. O diagrama a seguir, Figura 10, ilustra as relações em tempo de compilação e em tempo de execução dos arquivos de código-fonte em C#, as bibliotecas de classe do .NET Framework, *assemblies* e o CLR (MICROSOFT, 2015).

Figura 10 – Diagrama relação de compilação e execução em C#



Fonte: MICROSOFT, 2015.

2.3.3 Microsoft Office Excel e VBA

A plataforma utilizada para implementar os procedimentos de tratamento de dados foi o software Microsoft Excel, Figura 11, uma ferramenta para criação de planilhas eletrônicas pertence ao conjunto de aplicativos Office. O pacote de aplicativos do Office tem um sofisticado conjunto de recursos para criar, formatar e manipular documentos, e-mails, bancos de dados, formulários, planilhas e apresentações. A fim de automatizar tarefas repetitivas um dos recursos mais comuns do Office é o VBA (*Visual Basic Application*), recurso este utilizado neste trabalho para automatizar o processo de tratamento de dados (MICROSOFT, 2019).

Figura 11 – Microsoft Excel



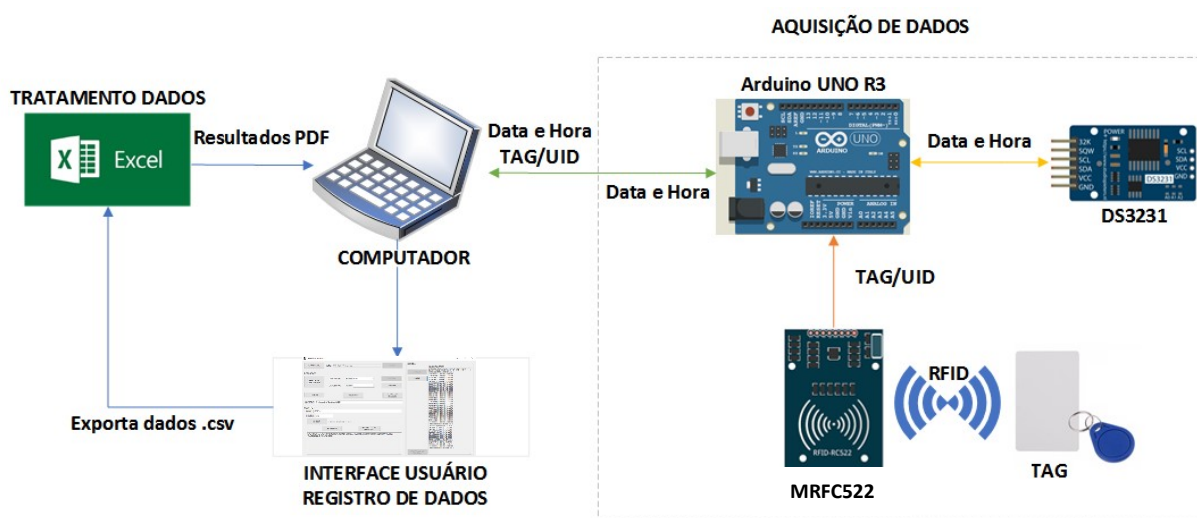
Fonte: AUTORIA PRÓPRIA, 2019.

3 - DESENVOLVIMENTO

3.1 Configurações gerais

Para este trabalho, foram esquematizadas configurações físicas, arquiteturas de comunicação, software para aquisição e registro de dados, e ferramentas para tratamento de dados, atendendo aos requisitos básicos para os testes, conforme indicado no diagrama da Figura 12.

Figura 12 – Diagrama geral



Fonte: AUTORIA PRÓPRIA, 2019.

Para a coleta dos dados utilizou-se módulos RFID-RC522 conectados a uma placa de prototipagem de utilização livre Arduino UNO, um módulo RTC DS3231 (*Real Time Clock Module* – Módulo de clock em tempo real) também foi conectado na placa para manter a sincronização de contagem de tempo.

Nos testes de comunicação e aquisição de dados, utilizou-se etiquetas RFID que ao entrar em contato com o campo eletromagnético gerado por um dos módulos RFID-RC522 conectados ao Arduino, cria uma diferença de potencial em sua bobina e alimenta o *transponder* que emite um sinal contendo seu código de identificação. Este código juntamente com uma marcação de tempo é transmitido via comunicação serial do módulo para o Arduino e também via comunicação serial do Arduino para o computador.

Um software com interface gráfica com o usuário do tipo Aplicação do Windows Forms, foi desenvolvido em linguagem C# no ambiente de desenvolvimento integrado de software da Microsoft, Microsoft Visual Studio. Este software é capaz de monitorar as portas

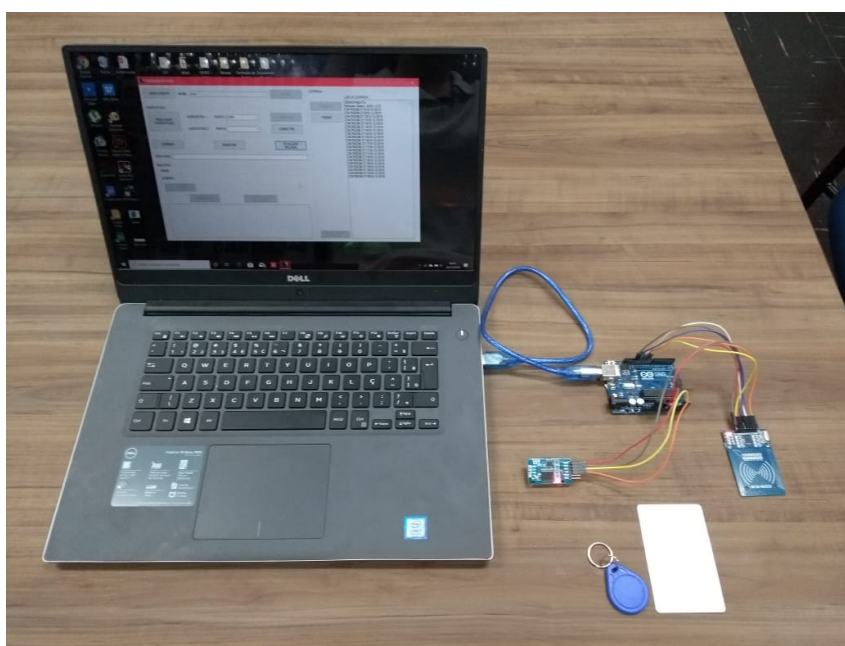
de comunicação serial do computador, coletar os dados enviados da placa Arduino e armazená-los em logs de dados, como registros de eventos.

O software também é capaz de criar eventos e nomeá-los para identificação dos registros de dados e credenciar as etiquetas RFID utilizadas. Os logs criados são salvos em um destino escolhido pelo usuário no computador. Uma planilha do Microsoft Excel com tabela dinâmica e botões para execução de comandos criados em VBA, foi desenvolvida para importar os logs, tratar os dados e organizá-los para fácil visualização pelo usuário.

3.2 Configurações físicas

Os dispositivos físicos utilizados neste trabalho foram: computador, cabo usb tipo B, placa de prototipagem Arduino Uno, módulo RTC DS3231 e módulos RFID RC522, conforme Figura 13.

Figura 13 – Dispositivos físicos

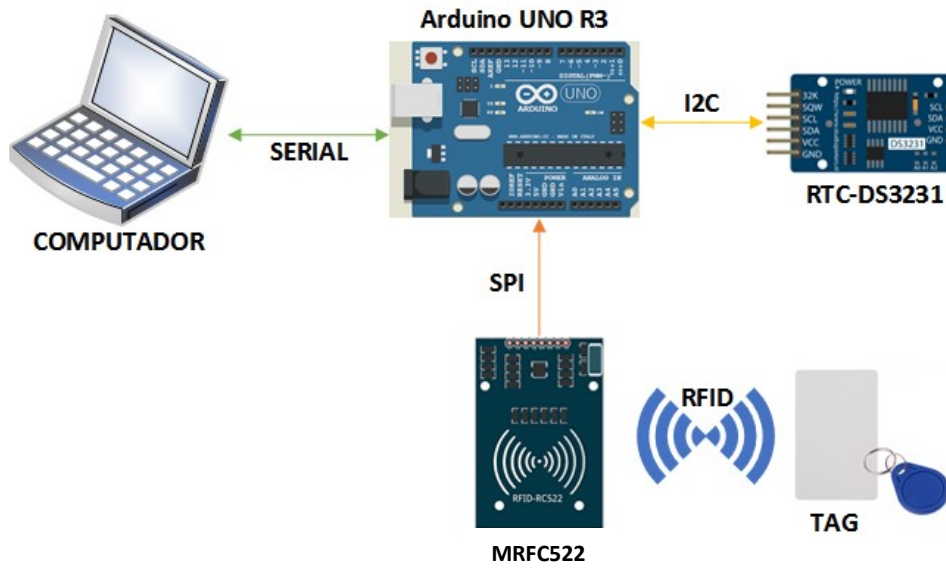


Fonte: AUTORIA PRÓPRIA, 2019.

3.3 Arquitetura de comunicação

A arquitetura de comunicação é representada de acordo com o diagrama na Figura 14. O computador e a placa Arduino UNO são conectados através de cabo USB, com terminais do tipo A para o computador e tipo B para a placa Arduino. O protocolo de comunicação utilizado é o serial RS232. Os módulos MRFC522 e RTC DS3231 são conectados a placa Arduino e comunicam-se via protocolo serial assíncrono SPI e I2C, respectivamente.

Figura 14 – Arquitetura de comunicação

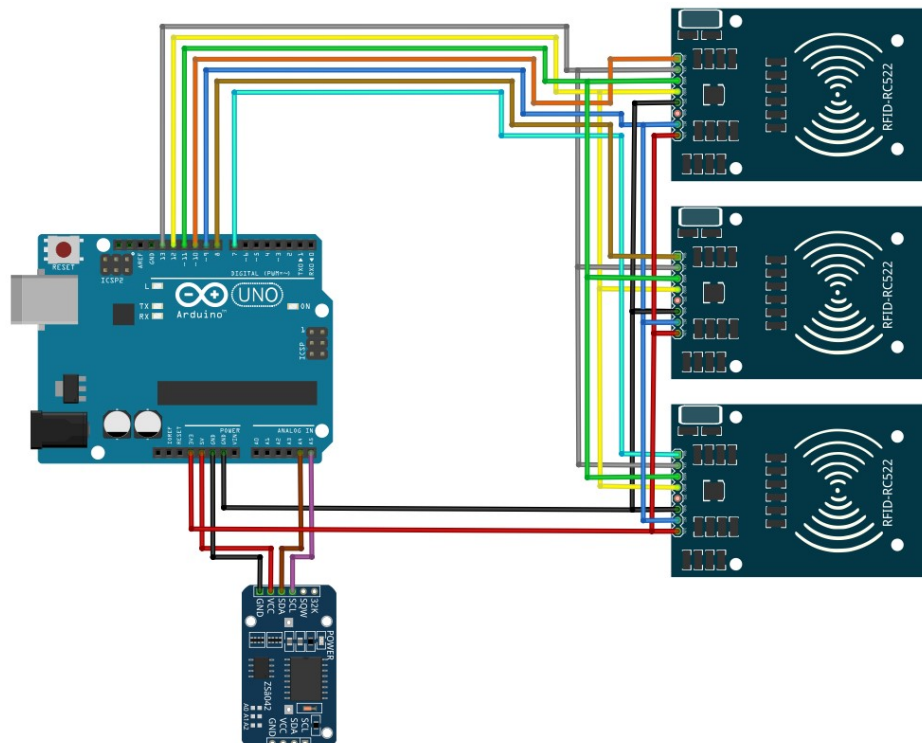


Fonte: AUTORIA PRÓPRIA, 2019.

3.4 Aquisição de dados

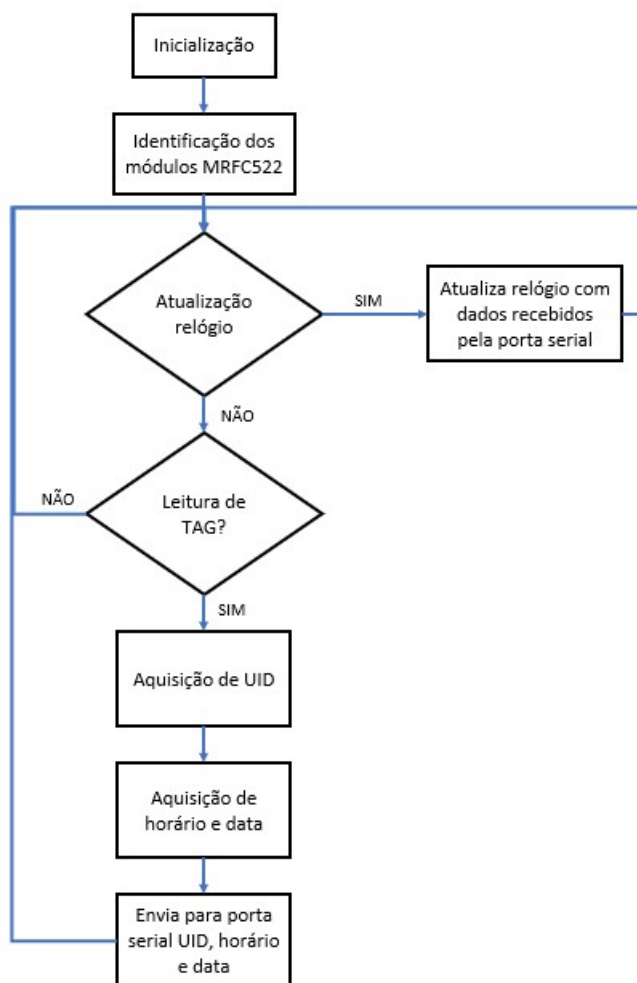
A aquisição de dados é realizada pelo conjunto placa Arduino UNO, módulo MRFC522 e módulo RTC DS3231, representados na Figura 15. Um fluxograma do funcionamento do código desenvolvido é exposto na Figura 16.

Figura 15 – Conjunto placa Arduino UNO, RFID-RC522 e RTC DS3231



Fonte: AUTORIA PRÓPRIA, 2019.

Figura 16 – Fluxograma código aquisição de dados



Fonte: AUTORIA PRÓPRIA, 2019.

De acordo com o código desenvolvido, primeiro é verificado a presença de uma *tag*, caso esteja presente os dados de sua memória são recebidos, decodificados e armazenados em um *buffer*, memória temporária. Uma subrotina é responsável por identificar o tamanho do dado armazenado, o percorrer e enviar para a porta serial os caracteres correspondentes a UID recebida. Posteriormente, outra subrotina recebe do módulo RTC DS3231 as informações de data e horário, do mesmo instante de leitura da *tag*.

O módulo RTC DS3231 possui uma bateria de 3 volts em sua placa de circuito, ou seja, alimentação independente da placa Arduino. Com essa propriedade ele é capaz de manter a contagem de *clock* em seu microprocessador, mesmo quando a placa Arduino UNO estiver desenergizada. Isto é um aspecto importante para a aquisição de dados neste trabalho, pois a leitura de horário é crucial para definir o tempo de prova dos candidatos em uma corrida.

Assim, ao utilizar vários conjuntos para a aquisição de dados, todos possuirão a mesma contagem de tempo. No entanto, em casos de sincronizar a contagem em conjuntos diferentes ou se a bateria de um módulo descarregar, é necessário atualizar a temporização do mesmo. Esta função é realizada pelo software CRONÔMETRO RFID, desenvolvido pelos autores deste trabalho, ao selecionar esta função do software, o horário e data do computador é lida e enviada via comunicação serial para o módulo.

Este dado possui uma formatação de texto específica, a qual uma subrotina do código desenvolvido para a placa Arduino identifica como comando para atualização de data e horário do módulo. Após a sincronização, a contagem atualizada do módulo é disposta para o usuário, para conferência e confirmação da atualização. Todas essas funções são dispostas em um loop, afim de manter a leitura constante do módulo RFID RC522 de novas *tags*.

3.5 Registro de dados

A aquisição dos dados transmitidos do Arduino para o computador e o registro dos mesmos, são realizados pelo software CRONÔMETRO RFID, Figura 17, desenvolvido pelos autores. O software possui interface gráfica para fácil manuseio por um usuário, suas ferramentas são: criar e nomear eventos; procurar dispositivos conectados via porta serial ao computador; estabelecer comunicação serial com o dispositivo conectado; atualizar contagem de data e horário; registrar nomes, números e *tags* ; registrar leituras de largada e chegada da corrida; escolher caminho para salvamento de logs; e salvar logs dos registros.

Figura 17 – Tela inicial CRONÔMETRO RFID

Fonte: AUTORIA PRÓPRIA, 2019.

O software é do tipo Aplicação do Windows Forms, desenvolvido em linguagem C# no Microsoft Visual Studio. As Figuras 18 e 19 demonstram algumas das abas de desenvolvimento do software CRONÔMETRO RFID, na Figura F é apresentada a edição do código principal e na Figura G a edição do design da interface gráfica. A figura 20 apresenta o fluxograma do código fonte.

Figura 18 – Edição código CRONÔMETRO RFID

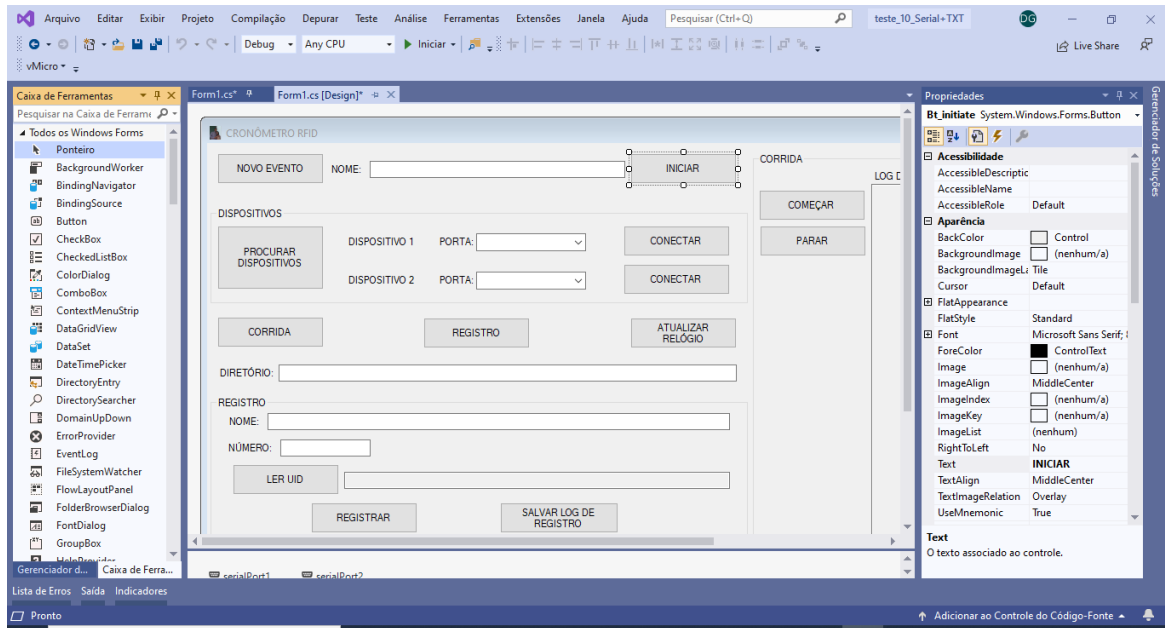
```

1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10 using System.IO.Ports;
11
12 namespace teste_10_Serial_TXT
13 {
14     public partial class Form1 : Form
15     {
16
17         public String in_data1;
18         public String in_data2;
19         public DateTime datetime;
20         private string evento;
21
22
23         1 referência
24         public Form1()
25         {
26             InitializeComponent();
27         }
28
29         1 referência
30         private void Bt_start_Click(object sender, EventArgs e)
31         {
32             Bt_stop.Enabled = true;
33         }
34     }
35 }

```

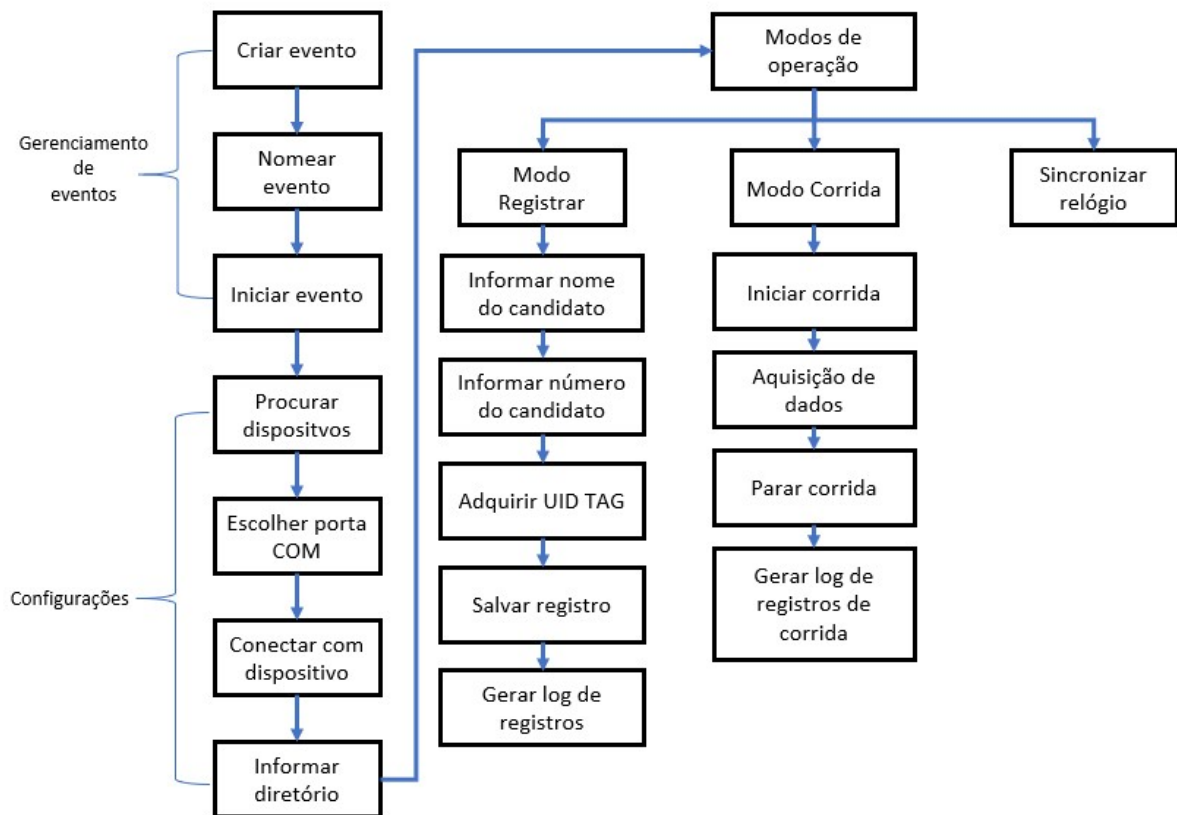
Fonte: AUTORIA PRÓPRIA, 2019.

Figura 19 – Edição interface gráfica CRONÔMETRO RFID



Fonte: AUTORIA PRÓPRIA, 2019.

Figura 20 – Fluxograma código fonte software CRONÔMETRO RFID



Fonte: AUTORIA PRÓPRIA, 2019.

A seguir são apresentadas as ferramentas e funcionalidades do software.

3.5.1 Criar eventos

O primeiro passo para utilização do software CRONÔMETRO RFID é criar um evento, isto é, nomear o acontecimento em que ocorrerá o registro dos dados, Figura 21, assim a leitura das etiquetas RFID é atrelada com a identificação do evento. Este passo é importante, pois com a nomeação do evento será possível a reutilização das etiquetas.

Figura 21 – Criar eventos

Fonte: AUTORIA PRÓPRIA, 2019.

Deve-se clicar em iniciar “NOVO EVENTO”, que habilita a aba nome, inserir uma identificação para o evento e clicar em “INICIAR”. O botão iniciar habilitará as opções de configuração e conexão de dispositivos.

3.5.2 Dispositivos

Após iniciado o evento caixa de configurações e conexão de dispositivos será habilitada, primeiro clica-se em “PROCURAR DISPOSITIVOS”, esta opção irá fazer uma varredura das portas seriais do computador e identificar em quais portas há dispositivos conectados, Figura 22.

Figura 22 – Dispositivos

The screenshot shows the 'CRONÓMETRO RFID' application window. At the top, there's a 'NOVO EVENTO' section with a 'NOME' field containing 'CORRIDA PARA O TCC' and an 'INICIAR' button. Below this is the 'DISPOSITIVOS' section, which includes a 'PROCURAR DISPOSITIVOS' button and two rows for 'DISPOSITIVO 1' and 'DISPOSITIVO 2'. Each row has a 'PORTA' dropdown menu. The first dropdown is open, showing 'COM8', 'COM7', and 'COM11'. There are 'CONECTAR' buttons for each device. In the center, there are three buttons: 'CORRIDA', 'REGISTRO', and 'ATUALIZAR RELÓGIO'. Below these are the 'REGISTRO' fields: 'DIRETÓRIO', 'NOME', and 'NÚMERO'. There are 'LER UID', 'REGISTRAR', and 'SALVAR LOG DE REGISTRO' buttons. On the right side, there's a 'CORRIDA' section with 'COMEÇAR' and 'PARAR' buttons, and a 'LOG DE CORRIDA' area. At the bottom right, there's a 'SALVAR LOG DE CORRIDA' button.

Fonte: AUTORIA PRÓPRIA, 2019.

Os dispositivos conectados via porta serial serão listados nas caixas de combinação “PORTA”, deve-se escolher aquela na qual a placa Arduino está conectada e clicar em “CONECTAR”, após serão habilitadas as opções de corrida, registro, sincronizar tempo e informar o diretório.

3.5.3 Diretório

Após conectar um dispositivo a caixa de texto “DIRETÓRIO”, Figura 23, é ativada.

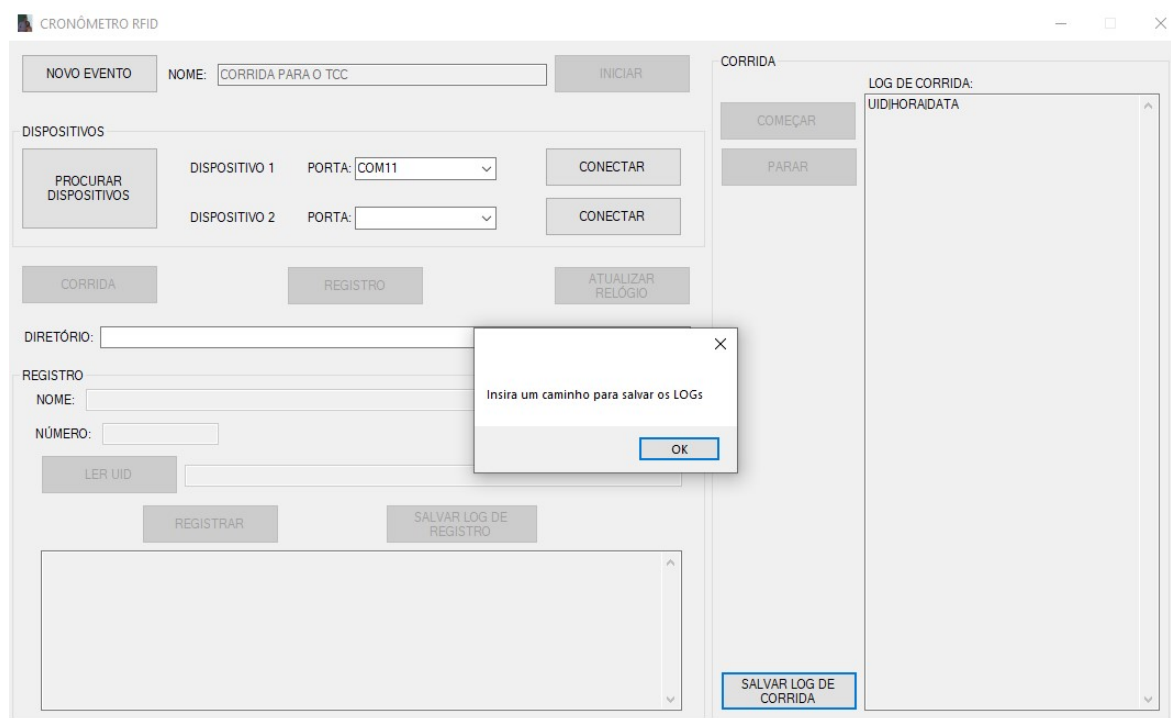
Figura 23 - Diretório

This screenshot shows the same 'CRONÓMETRO RFID' application window. The 'DIRETÓRIO' field is now populated with the text 'C:\Users\d_pir\Desktop\DATA'. The 'DISPOSITIVOS' section shows that 'COM11' is selected in the 'PORTA' dropdown for 'DISPOSITIVO 1'. The 'CORRIDA' section has 'COMEÇAR' and 'PARAR' buttons. The 'REGISTRO' section has 'REGISTRAR' and 'SALVAR LOG DE REGISTRO' buttons. The 'LOG DE CORRIDA' area is empty.

Fonte: AUTORIA PRÓPRIA, 2019.

Com ela é possível inserir o caminho para salvar no computador os logs de registros e de leituras de corrida, caso esta caixa de texto não esteja preenchida com um endereço uma mensagem será exibida, informando que o caminho não foi inserido, Figura 24.

Figura 24 – Mensagem diretório vazio

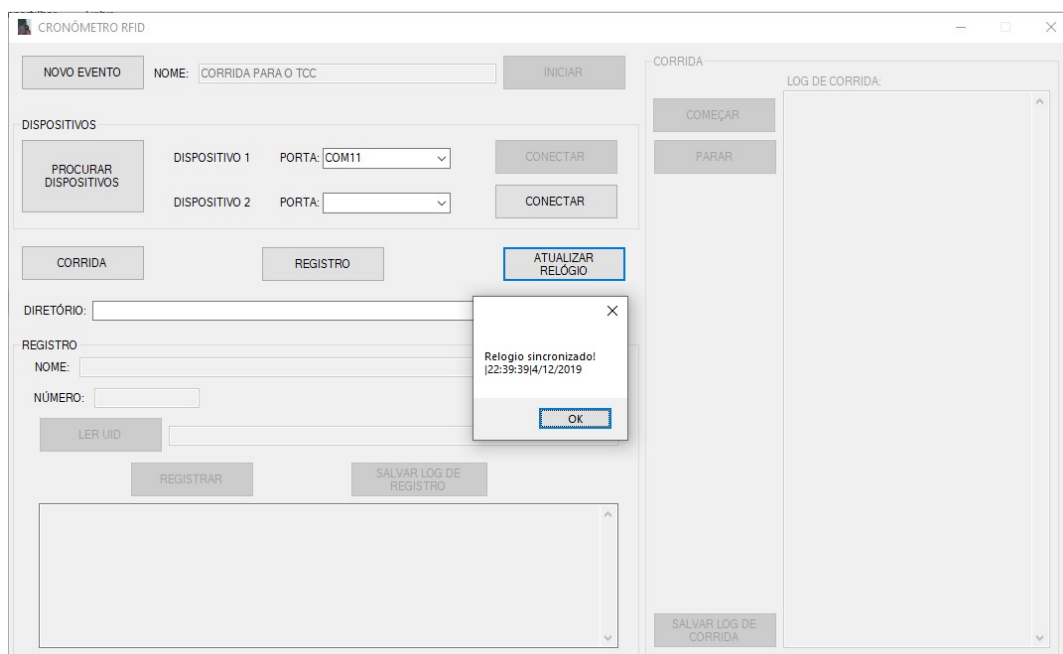


Fonte: AUTORIA PRÓPRIA, 2019.

3.5.4 Sincronizar Tempo

O módulo RTC DS3231 conectado ao Arduino é responsável pela contagem de data e horário, mesmo quando a placa Arduino estiver desenergizada pois ela possui uma bateria para alimentação independente, no entanto decorrido grande período de tempo esta bateria pode se esgotar e interromper a contagem de *clock* do módulo, ou em casos de utilização de mais placas Arduino com os módulos para aquisição de dados, a contagem dos módulos RTC DS3231 em cada placa podem estar dessincronizadas. Por isso é importante utilizar esta ferramenta que sincroniza a contagem do módulo com a data e horário real do computador, Figura 25, isto é feito por meio do envio das informações adquiridas pelo software do computador para a placa Arduino, com um formato de texto específico. Esta informação será recebida como um comando para que o módulo atualize sua contagem com a data e horário recebidos.

Figura 25 – Sincronizar tempo



Fonte: AUTORIA PRÓPRIA, 2019.

Feita a sincronização do relógio uma caixa de mensagem é exibida informando que a atualização foi efetuada e a data e horário lido no mesmo instante.

3.5.5 Modo Registro

O modo “REGISTRO”, Figura 26, permite que o usuário cadastre no software os candidatos da prova, informando nome, número e UID da *tag*, para ler a UID deve-se aproximar a *tag* do módulo MRFC522 que irá efetuar a leitura e será preenchido automaticamente o campo. Com estes três dados preenchidos pode-se clicar no botão “REGISTRAR” que irá incluir os dados no log de registros.

Figura 26 – Modo Registro

Fonte: AUTORIA PRÓPRIA, 2019.

Com os registros efetuados é possível salvar o log de registro clicando no botão “SALVAR LOG DE REGISTRO”, Figura 27, que irá agrupar os registros efetuados em um arquivo do tipo .csv no local escolhido pelo usuário.

Figura 27 – SALVAR LOG DE REGISTROS

Fonte: AUTORIA PRÓPRIA, 2019.

3.5.6 Modo Corrida

Ao escolher o modo corrida, Figura 28, irá ativar a opção de começar uma corrida, botão “COMEÇAR”. Ao pressionar este botão iniciará a leitura constante dos dados enviados do módulo MRFC522 pela porta serial ao computador, assim serão lidas todas as etiquetas que passarem pelos módulos nas linhas de saída e de chegada. Estas leituras serão salvas no log de corrida.

Figura 28 – Modo Corrida

The screenshot shows the 'CRONÔMETRO RFID' application window. The main interface is divided into several functional areas:

- NOVO EVENTO:** A section for creating a new event, with a text field containing 'CORRIDA PARA O TCC' and an 'INICIAR' button.
- DISPOSITIVOS:** A section for selecting devices, with two 'DISPOSITIVO' fields (one set to 'COM11') and 'CONECTAR' buttons.
- CORRIDA:** A section for starting and stopping the race, featuring 'COMEÇAR' and 'PARAR' buttons.
- REGISTRO:** A section for recording data, with fields for 'NOME' (LETICIA), 'NÚMERO' (456), and 'LER UID' (C547692322:42:6A4/12/2019), along with 'REGISTRAR' and 'SALVAR LOG DE REGISTRO' buttons.
- LOG DE CORRIDA:** A section displaying a list of race log entries, with a 'SALVAR LOG DE CORRIDA' button below it.

The log entries in the bottom right corner include the following data points:

```

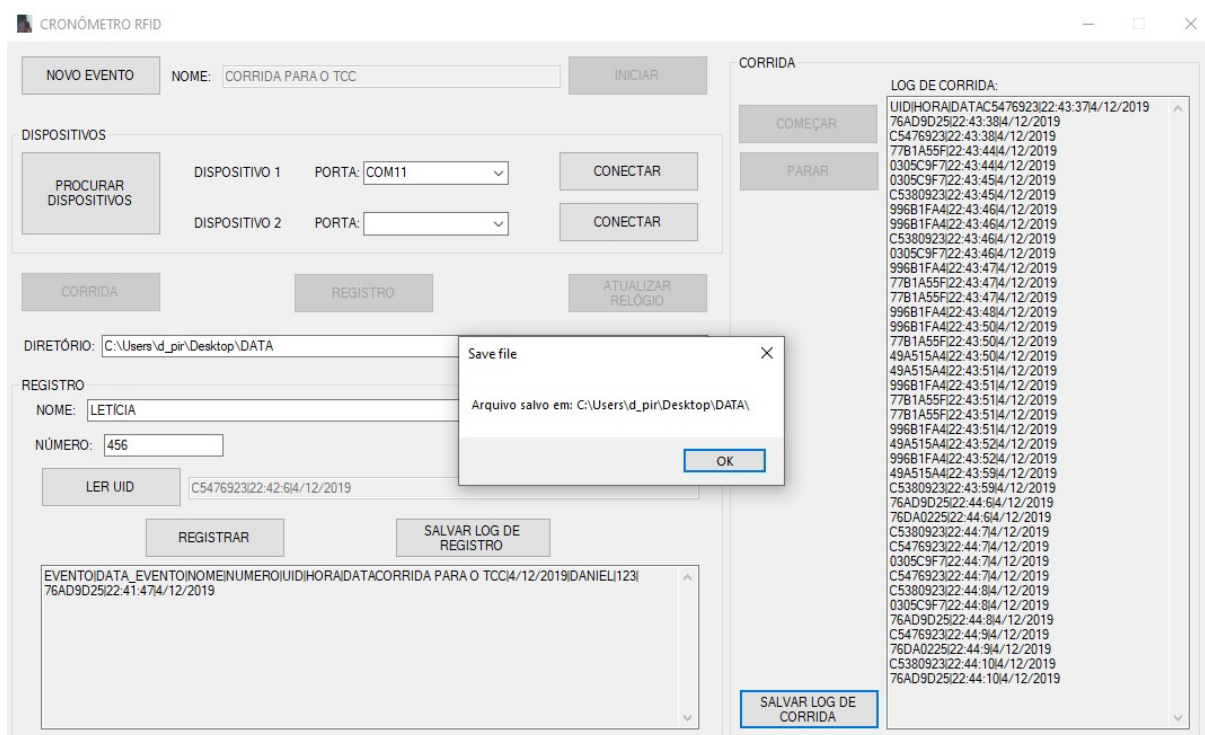
UID|HORADATA|C547692322:43:374/12/2019
76AD9D2522:43:384/12/2019
C547692322:43:384/12/2019
77B1A55F22:43:444/12/2019
0305C9F722:43:444/12/2019
0305C9F722:43:454/12/2019
C538092322:43:464/12/2019
996B1FA422:43:464/12/2019
996B1FA422:43:464/12/2019
C538092322:43:464/12/2019
0305C9F722:43:464/12/2019
996B1FA422:43:474/12/2019
77B1A55F22:43:474/12/2019
77B1A55F22:43:474/12/2019
996B1FA422:43:484/12/2019
996B1FA422:43:504/12/2019
77B1A55F22:43:504/12/2019
49A515A422:43:504/12/2019
49A515A422:43:514/12/2019
996B1FA422:43:514/12/2019
77B1A55F22:43:514/12/2019
77B1A55F22:43:514/12/2019
996B1FA422:43:514/12/2019
49A515A422:43:524/12/2019
996B1FA422:43:524/12/2019
49A515A422:43:594/12/2019
C538092322:43:594/12/2019
76AD9D2522:44:6A4/12/2019
76D022522:44:6A4/12/2019
C538092322:44:7A4/12/2019
C547692322:44:7A4/12/2019
C538092322:44:8A4/12/2019
0305C9F722:44:8A4/12/2019
76AD9D2522:44:8A4/12/2019
C547692322:44:9A4/12/2019
76D022522:44:9A4/12/2019
C538092322:44:104/12/2019
76AD9D2522:44:104/12/2019

```

Fonte: AUTORIA PRÓPRIA, 2019.

Ao terminar a corrida deve-se clicar no botão “PARAR”, esta opção irá interromper o registro de novas *tags*, pois irá fechar a comunicação via porta serial com o dispositivo. Será ativado o botão “SALVAR LOG DE CORRIDA”, Figura 29, que permite exportar o log de leituras da corrida em arquivo do tipo .csv para o local informado pelo usuário, após clicar neste botão uma mensagem confirmação do salvamento será exibida informando também o caminho onde o arquivo foi salvo.

Figura 29 – SALVAR LOG DE CORRIDA



Fonte: AUTORIA PRÓPRIA, 2019.

Ao encerrar a aquisição de dados, após pressionar o botão “PARAR”, a comunicação via porta serial com o dispositivo é fechada. Para iniciar nova aquisição de dados é necessário pressionar o botão “CONECTAR” nas opções de dispositivo novamente. Assim a comunicação via porta serial com o dispositivo conectado será reestabelecida.

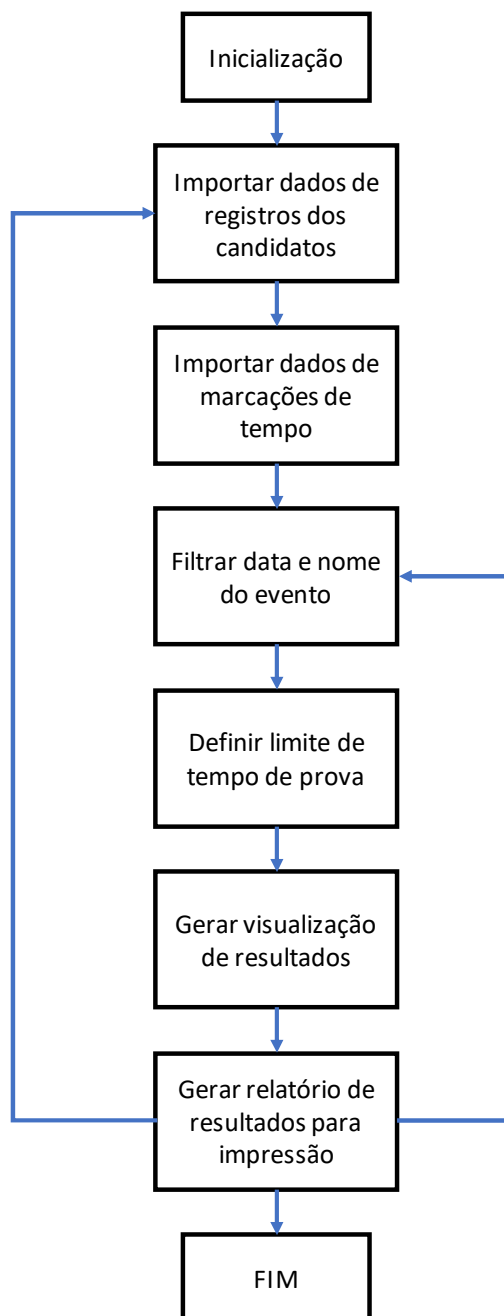
3.6 Tratamento de dados no Excel

O tratamento dos dados salvos em arquivo formato . csv é realizado em uma planilha de Excel, programada com códigos em VBA , tabela dinâmica e fórmulas essenciais. O processo de tratamento de dados ocorre em 6 etapas:

- Passo 1: Importar os dados de registro;
- Passo 2: Importar os dados de corrida;
- Passo 3: Filtrar os dados desejados;
- Passo 4: Determinar o limite de tempo de prova;
- Passo 5: Gerar a lista de resultados;
- Passo 6: Gerar a folha de resultados para impressão.

A Figura 30 apresenta o fluxograma do código fonte para o tratamento de dados na planilha.

Figura 30 – Fluxograma código fonte planilha de tratamento de dados



Fonte: AUTORIA PRÓPRIA, 2019.

Uma aba principal é exibida para o usuário, Figura 31, nela constam os botões “IMPORTAR DADOS DE REGISTRO”, “IMPORTAR DADOS DE CORRIDA”, “GERAR LISTA DE RESULTADOS”, “GERAR FOLHA DE RESULTADO”, as opções de filtro para data e nome do evento, campo para informar o tempo de prova, e tabela com os nomes dos candidatos, tempos de prova de cada candidato, classificação e resultado de seu desempenho.

Figura 31 – Aba de interface com o usuário



Fonte: AUTORIA PRÓPRIA, 2019.

3.6.1 Importar dados

Os botões “IMPORTAR DADOS DE REGISTRO” e “IMPORTAR DADOS DE CORRIDA” permitem ao usuário importar os arquivos de logs para a planilha, durante os processos de importação são utilizadas outras três abas, uma é um banco de dados que armazena todos os registros importados para a planilha, e duas são abas auxiliares para formatar conteúdo e aplicar fórmulas. Primeiro deve-se importar o log de dados de registro, estes são os dados cadastrados dos competidores ou candidatos nas provas de corrida, esses dados são trabalhados com os comandos do código em VBA na aba “aux_r” e carregados na aba “BD” (Banco de Dados), Figura 32.

Figura 32 – Banco de dados após carregar dados de registros

	A	B	C	D	E	F	G
	EVENTO	DATA	NOME	N° CANDIDATO	UID	HORA CADASTRO	DATA CADASTRO
1	ESTE_CABULOS	12/04/2019	CABULOS1	12	C58E1C23	18:46:05	12/04/2019
2	ESTE_CABULOS	12/04/2019	CABULOSO2	46	C5476923	18:46:21	12/04/2019
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							

Fonte: AUTORIA PRÓPRIA, 2019.

Após importar os dados de registro é possível importar os dados de corrida clicando o botão “IMPORTAR DADOS DE CORRIDA”, os dados importados são tratados na aba “aux_c”, Figura 33, aba auxiliar para tratar os dados de corrida, os comandos do código VBA aplica alterações e fórmulas para preencher os campos remanescentes da tabela do banco de dados.

Figura 33 – Banco de dados após carregar dados de corrida

	A	B	C	D	E	F	G	H	I	J	K	L
1	EVENTO	DATA	NOME	N° CANDIDATO	UID	HORA CADASTRO	DATA CADASTRO	EVENTO&UID	TEMPO INICIAL	tf	TEMPO FINAL	TEMPO
2	ESTE_CABULOS	12/04/2019	CABULOS1	12	C58E1C23	18:46:05	12/04/2019	CABULOSOC5	18:46:36	JULOSOC	18:47:53	00:01:17
3	ESTE_CABULOS	12/04/2019	CABULOSO2	46	C5476923	18:46:21	12/04/2019	CABULOSOC5	18:46:41	JULOSOC	18:47:51	00:01:10
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												
28												
29												
30												
31												

Fonte: AUTORIA PRÓPRIA, 2019.

3.6.2 Listas de resultados

Após a atualização do banco de dados, as informações do novo evento importado estarão disponíveis nos campos de filtragem, Figura 34.

Figura 34 – Campos de filtros

EVENTO	DATA
CORRIDA PARA O TCC	12/04/2019
	15/02/1900
	13/05/1950
	02/05/1963
	13/05/2018

Fonte: AUTORIA PRÓPRIA, 2019.

Para gerar uma lista, o usuário deve escolher uma opção de evento e data, informar o tempo limite de prova, Figura 35, para avaliar o desempenho do candidato e clicar no botão “GERAR LISTA DE RESULTADO”.

Figura 35 – Tempo de prova e botão gerar lista de resultado

Passo 1: Importe os dados de registro
 Passo 2: Importe os dados de corrida
 Passo 3: Filtre os dados desejados
 Passo 4: Determine o limite de tempo de prova
 Passo 5: Gere a lista de resultados
 Passo 6: Gere a folha de resultados para impressão

Evento: CORRIDA PARA O TCC
 Data: 12/04/2019

Tempo de Prova: 00:15:00

Nome	Tempo	Resultado
1 DANIEL	00:00:28	APROVADO

Fonte: AUTORIA PRÓPRIA, 2019.

A lista gerada, Figura 36, será exibida na tela inicial, contendo nome e data do evento, classificação, nome, tempo de prova e resultado do candidato.

Figura 36 – Tempo de prova e botão gerar lista de resultado

CRONÔMETRO RFID

Evento: CORRIDA PARA O TCC
 Data: 12/04/2019

Nome	Tempo	Resultado
1 DANIEL	00:00:28	APROVADO

EVENTO
 CORRIDA PAI

Fonte: AUTORIA PRÓPRIA, 2019.

Com a opção “GERAR FOLHA DE RESULTADO” o programa formata um design de página padrão, Figura 37, para exportar a lista de resultado em arquivos do tipo PDF (*Portable Document Format*) ou para impressão.

Figura 37 – Página padrão para exportar ou imprimir lista de resultado

UFG

RESULTADO TESTE DE APTIDÃO FÍSICA

Centro de Seleção UFG

PROVA: CORRIDA PARA O TCC
 DATA: 12/04/2019

Nome	Tempo	Resultado
1 DANIEL	00:00:28	APROVADO

Fonte: AUTORIA PRÓPRIA, 2019.

4 – RESULTADOS E DISCUSSÃO

O desenvolvimento de produtos baseados no design interativo é uma abordagem amplamente utilizada por várias empresas. O projeto iterativo é o processo de desenvolvimento de novos produtos através de uma série de etapas planejadas, onde cada etapa é marcada pelo desenvolvimento de uma nova versão do produto, chamada de protótipo. As duas versões de protótipos mais importantes são as denominadas “alfa” e “beta” (MARESPIERA, 2019).

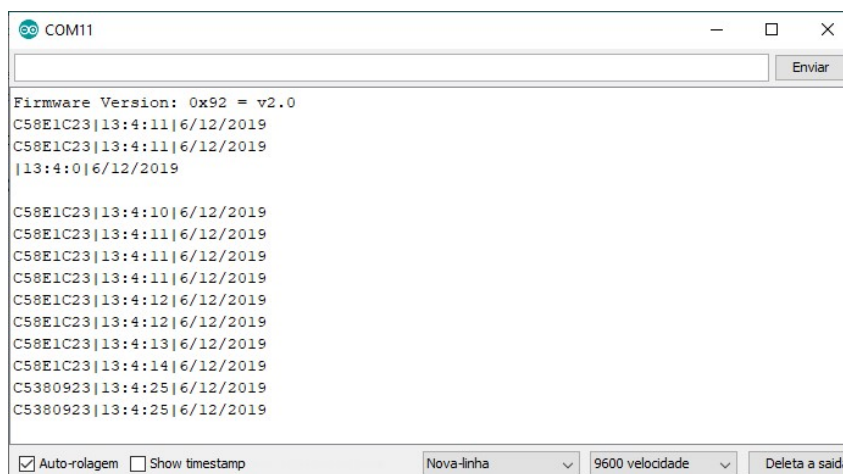
O processo de desenvolvimento se inicia na fase de “prova de conceito”, é neste ponto que se determina se a ideia do produto é viável. Caso a ideia possua um potencial positivo, o protótipo alfa é construído. Este não se destina a ser totalmente funcional, em vez disso, seu objetivo é servir como fins de teste para definir a direção dos próximos passos de desenvolvimento, eliminando assim as características impraticáveis do projeto. Os aspectos positivos deste estágio são passados para a próxima fase, conhecida como fase beta (MARESPIERA, 2019).

A fase de protótipo beta, embora não perfeitamente funcional e pronta para produção, é a versão mais próxima do produto final, geralmente alguns erros e questões de design ainda estão pendentes neste ponto. No entanto, o protótipo beta pode ser utilizado em situações reais para testes, a fim de identificar possíveis falhas. Os problemas encontrados e ainda pendentes são analisados e tratados até que um protótipo beta totalmente funcional seja fabricado, este então é usado como base para a fase final de desenvolvimento (MARESPIERA, 2019).

A ideia conceito para o produto foi confirmada, o que prova a possibilidade de utilização da tecnologia de identificação por rádio frequência (RFID). Os processos de aquisição, registro e tratamento de dados, que são os principais aspectos do trabalho foram positivos. Como resultado tem-se a leitura de *tags* de identificação, reconhecimento de identificação da *tag*, horário de leitura, cadastro de informações em banco de dados, transmissão, tratamento e exibição de dados em interfaces gráficas de fácil utilização.

Para a etapa de aquisição de dados um conjunto composto por placa Arduino UNO, módulo MRFC522 e módulo RTC DS3231 foi composto. Na Figura 38 comprova-se a leitura e identificação das *tags*, determinação de horário de leitura e envio dos dados via protocolo de comunicação serial, estes dados estão expostos no monitor de comunicação serial da plataforma de desenvolvimento IDE Arduino.

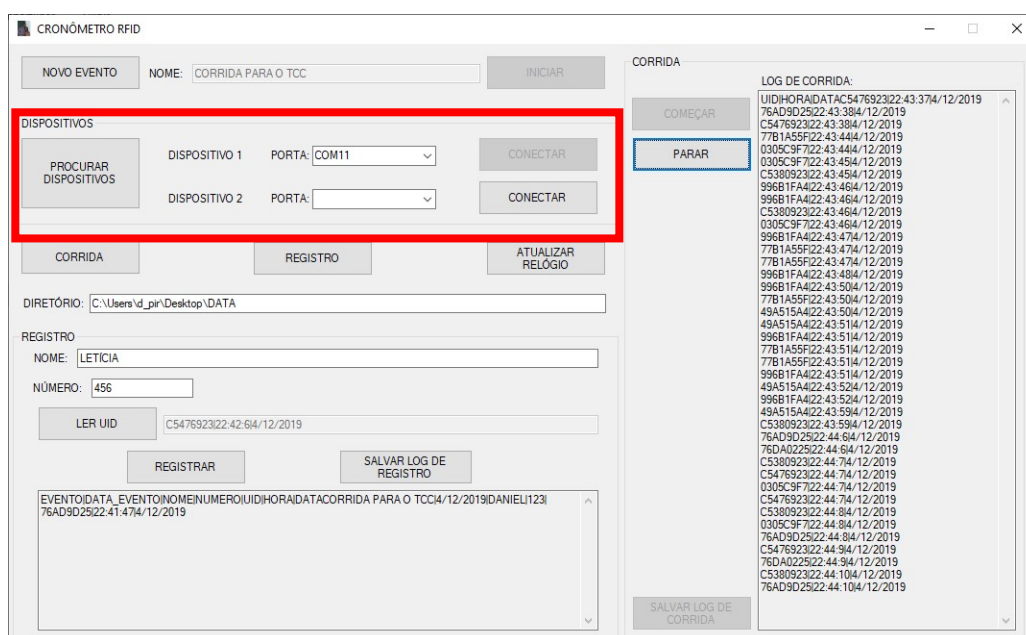
Figura 38 – Conjunto placa Arduino UNO, RFID-RC522 e RTC DS3231



Fonte: AUTORIA PRÓPRIA, 2019.

A etapa de registro de dados é realizada pelo software CRONÔMETRO RFID, desenvolvido pelos autores, desenvolvido na linguagem de programação C# no ambiente de desenvolvimento de softwares Microsoft Visual Studio. O software possui as funções de criar e nomear eventos, procurar dispositivos conectados via porta serial ao computador, estabelecer comunicação serial com o dispositivo conectado, atualizar contagem de data e horário, registrar nomes, números e *tags*, registrar leituras de largada e chegada da corrida, escolher caminho para salvamento de logs e salvar logs dos registros. O Software conecta-se ao conjunto de aquisição de dados via protocolo de comunicação serial, na Figura 39 demonstra-se a conexão com o dispositivo.

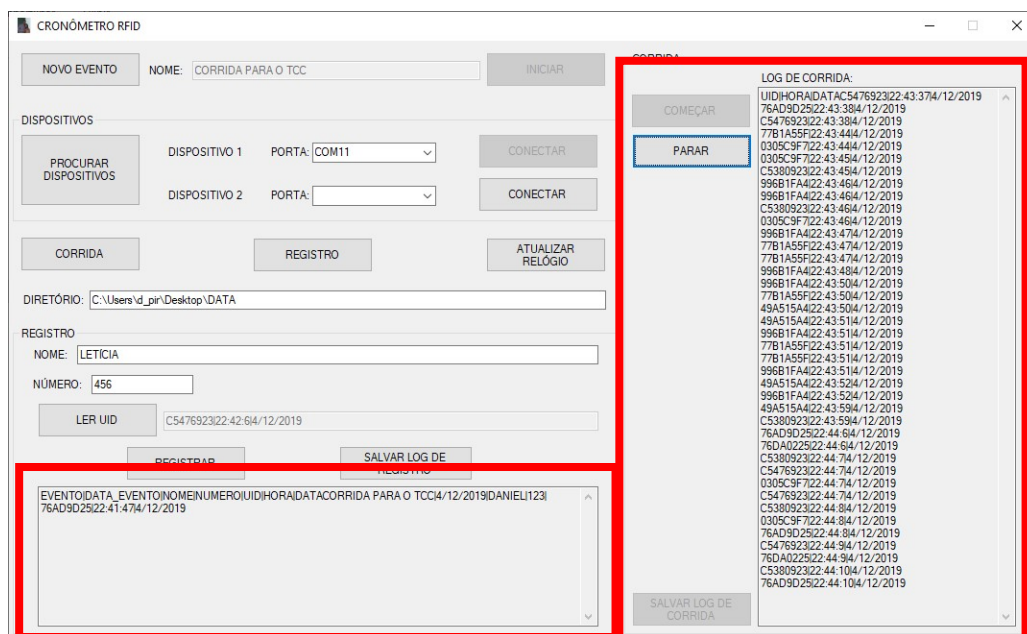
Figura 39 – Estabelecimento de comunicação serial



Fonte: AUTORIA PRÓPRIA, 2019.

A Figura 40 expõe as funções de registro de candidatos para prova de corrida e registro de aquisição de dados.

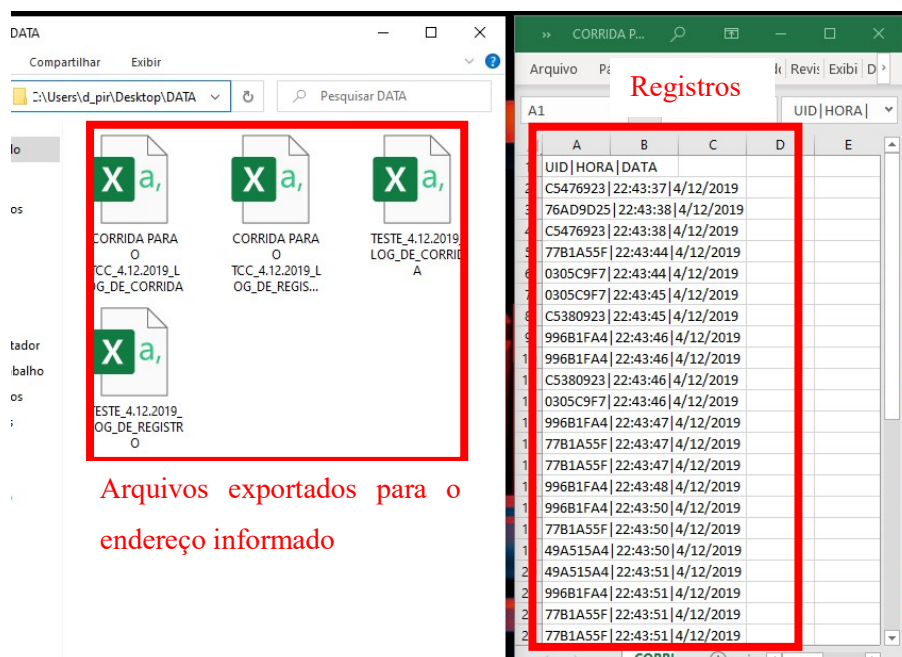
Figura 40 – Registro de candidatos para prova de corrida e de aquisição de dados



Fonte: AUTORIA PRÓPRIA, 2019.

Com as funções “SALVAR LOG DE DADOS” e “SALVAR LOG DE CORRIDA” os dados registrados no software podem ser exportados em arquivo do formato CSV e salvos na memória do computador no local selecionado pelo usuário. A Figura 41 expõe os arquivos exportados.

Figura 41 – arquivos exportados e registros



Arquivos exportados para o endereço informado

Fonte: AUTORIA PRÓPRIA, 2019.

A etapa de tratamento de dados é realizada em uma planilha do Excel, projetada com códigos em VBA, design, fórmulas especiais, e interface de fácil utilização. Na Figura 42 expõe-se a visualização dos dados tratados de evento teste.

Figura 42 – arquivos exportados e registros

CRONÔMETRO RFID			
IMPORTAR DADOS REGISTRO	Visualização de dados	GERAR FOLHA DE RESULTADO	
Evento: TCC	Data: 12/06/2019		
	Nome	Tempo	Resultado
1	Little John	00:05:43	APROVADO
2	Einstein	00:05:43	APROVADO
3	Edimilson Faria	00:05:43	APROVADO
4	Without Arms	00:05:44	APROVADO
5	Manuel de Ramos	00:05:44	APROVADO
6	Maria	00:05:45	APROVADO

Fonte: AUTORIA PRÓPRIA, 2019.

Os dados tratados de um evento selecionado podem ser organizados em uma página formatada para impressão ou gerar arquivo no formato PDF, como o exemplo da Figura 43.

Figura 43 – arquivos exportados e registros

UFG		RESULTADO TESTE DE APTIDÃO FÍSICA		Centro de Seleção UFG	
PRONIA: TCC	DATA: 12/06/2019				
	Nome	Tempo	Resultado		
1	Little John	00:05:43	APROVADO		
2	Einstein	00:05:43	APROVADO		
3	Edimilson Faria	00:05:43	APROVADO		
4	Without Arms	00:05:44	APROVADO		
5	Manuel de Ramos	00:05:44	APROVADO		
6	Maria	00:05:45	APROVADO		

Fonte: AUTORIA PRÓPRIA, 2019.

Para chegar até esta solução apresentado foi demandada grande quantidade de tempo voltada para pesquisas bibliográficas, desenvolvimento e testes. Somadas as várias tentativas que não obtiveram êxito, fazendo voltar a etapa de pesquisas. As etapas de registro de dados e tratamento de dados, segundo a analogia anteriormente citada, obtiveram resultados a nível de protótipo beta, ambas poderão ser utilizadas em testes reais. No entanto, a etapa de aquisição de dados pode ser classificada como protótipo nível alfa, pois o conceito de desenvolvimento foi provado, porém o alcance de leitura do módulo MFRC522 é curto, próximo a 10 cm, o que invalida sua utilização em testes reais para aquisição de dados.

Diante dessa problemática buscou-se soluções alternativas para aumentar o alcance de leitura das *tags*. As tentativas iniciais se referem a alterações no próprio módulo, como aumentar o ganho de sinal e confeccionar uma nova antena. A alteração no ganho de sinal do módulo é uma etapa relativamente simples, pois consiste apenas em alterações no código de biblioteca do módulo na plataforma IDE Arduino, conforme indica a Figura 44. Na função de iniciar a leitura do módulo é definido o ganho de sinal, o que possibilita alterá-lo e modificar para o ganho máximo. Com a alteração de ganho de sinal definida em máxima o alcance de leitura do módulo aumentou apenas poucos centímetros, entre 2 a 3 cm.

Figura 44 – Alterações ganho de sinal módulo MFRC 522

```

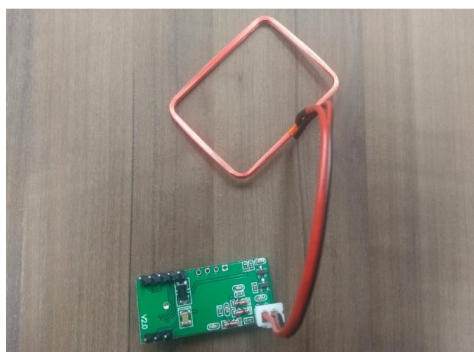
// When communicating with a PICC we need a timeout if something goes wrong.
// f_timer = 13.56 MHz / (2*TPreScaler+1) where TPreScaler = [TPrescaler_Hi:TPrescaler_Lo].
// TPrescaler_Hi are the four low bits in TModeReg. TPrescaler_Lo is TPrescalerReg.
PCD_WriteRegister(TModeReg, 0x80); // TAuto=1; timer starts automatically at
PCD_WriteRegister(TPrescalerReg, 0xA9); // TPreScaler = TModeReg[3..0]:TPrescalerReg, ie
PCD_WriteRegister(TReloadRegH, 0x03); // Reload timer with 0x3E8 = 1000, ie 25ms before
PCD_WriteRegister(TReloadRegL, 0xE8);
PCD_WriteRegister(TxASKReg, 0x40); // Default 0x00. Force a 100 % ASK modulation ind
PCD_WriteRegister(ModeReg, 0x3D); // Default 0x3F. Set the preset value for the CRC
PCD_WriteRegister(RFCfgReg, (0x07<<4)); // Set Rx Gain to max
PCD_AntennaOn(); // Enable the antenna driver pins
} // End PCD_Init()

```

Fonte: AUTORIA PRÓPRIA, 2019.

A segunda tentativa foi utilizar o módulo RFID RDM6300, Figura 45, este módulo opera na faixa de frequência de 125 kHz.

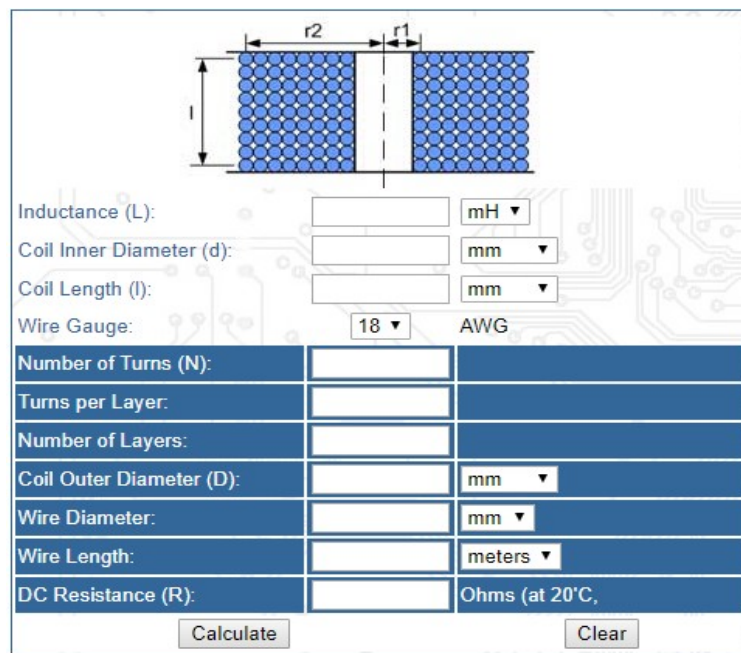
Figura 45 – Alterações ganho de sinal módulo MFRC 522



Fonte: AUTORIA PRÓPRIA, 2019.

No entanto, o alcance máximo medido deste módulo foi inferior ao MFRC522, próximo a 5 centímetros. Para reverter este problema tentou-se confeccionar uma nova antena para o módulo. Conforme algumas pesquisas indicaram a possibilidade de aumentar o alcance de sinal substancialmente. Para projetar a nova antena do tipo bobina de várias camadas com núcleo de ar, utilizou-se uma calculadora, Figura 46, disponível no site “*All about electronic circuits*” para definir os seus parâmetros.

Figura 46 – Calculadora de parâmetros para bobina de várias camadas com núcleo de ar



The image shows a calculator interface for a multi-layer air-core coil. At the top, a diagram illustrates a coil with an inner radius r_1 , an outer radius r_2 , and a total length l . The coil is composed of multiple layers of wire, represented by blue circles. Below the diagram, the calculator has several input fields and buttons:

- Inductance (L): mH
- Coil Inner Diameter (d): mm
- Coil Length (l): mm
- Wire Gauge: AWG
- Number of Turns (N):
- Turns per Layer:
- Number of Layers:
- Coil Outer Diameter (D): mm
- Wire Diameter: mm
- Wire Length: meters
- DC Resistance (R): Ohms (at 20°C)

Buttons for "Calculate" and "Clear" are located at the bottom of the interface.

Fonte: ALL ABOUT ELETRONIC CIRCUITS, 2010.

A calculadora baseia-se na fórmula (1), desenvolvida por Harold Alden Wheeler, para calcular indutâncias aproximadas em bobinas de várias configurações.

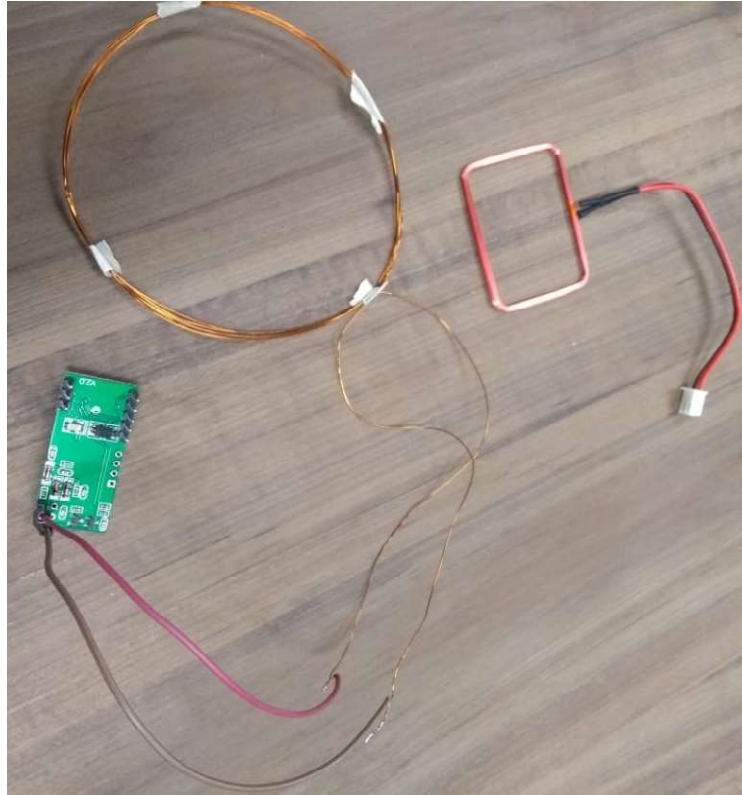
$$L(\mu H) = \frac{31,6 * N^2 * r_1^2}{6 * r_1} + 9 * l + 10 * (r_2 - r_1) \quad (1)$$

Onde $L(\mu H)$ é o valor de indutância em micro Henry, N é o número total de voltas, r_1 é o raio do interior da bobina, r_2 é raio externo da bobina e l é o comprimento total da bobina, ambos em metros (WHEELER, 1928).

Com este cálculo, busca-se identificar o número de voltas na bobina, para as dimensões e valor de indutância estabelecido, de acordo com as especificações técnicas do módulo, a indutância da antena está entre 47 a 68 μH . Os resultados com a nova antena,

Figura 47, não foram satisfatórios, pois seu alcance permaneceu semelhante a antena original do módulo.

Figura 47 – Calculadora de bobina com núcleo de ar de várias camadas



Fonte: AUTORIA PRÓPRIA, 2019.

Portanto, as tentativas para aumentar o alcance para aquisição de dados foram insatisfatórias, comprovando apenas a aplicabilidade de identificação de *tags* por rádio frequência. Para reverter este problema a solução é utilizar outras antenas de maior porte que operam nas faixas de alta frequência e ultra alta frequência.

5 - CONCLUSÃO

Este Projeto de Final de Curso buscou proporcionar a oportunidade de desenvolvimento de um protótipo de sistema para solucionar um problema enfrentado pelo Centro de Seleção da Universidade Federal de Goiás (CS - UFG), ao aplicar provas de corrida em Testes de Aptidão Física (TAF) de concursos públicos. O atual método de cronometragem utilizado é o manual, e devido a grande quantidade de candidatos necessita de muitas de pessoas para realizar a tarefa.

O objetivo do trabalho foi desenvolver um protótipo de sistema automatizado para cronometragem de provas de corridas. O sistema deveria ser capaz de aumentar a precisão na cronometragem, facilitar o gerenciamento de candidatos, aquisição de dados e divulgação dos relatórios de resultados. Ou seja, garantir maior segurança, reduzir custos, probabilidade de erros por fator humano e agilizar a aplicação desta etapa do TAF nos concursos realizados pelo CS - UFG.

Foi considerada a metodologia de projeto iterativo, na qual os resultados deste trabalho foram classificados em níveis de desenvolvimento, onde avalia-se a evolução de cada fase. Como as etapas de protótipos alfa e beta, as quais são fundamentais no aprimoramento. Esse trabalho possui três partes essenciais, são elas: aquisição de dados, registro de dados e tratamento de dados.

As etapas de registro de dados e tratamento de dados obtiveram resultados que as classificam como protótipo nível beta, pois podem ser utilizadas em testes reais. Já a etapa de aquisição de dados foi considerada no nível de protótipo alfa, nível este que certifica a aplicabilidade do conceito desenvolvido, porém não está apto a participar de testes reais. Esta etapa foi classificada como nível alfa, pois o alcance de leitura do módulo utilizado não foi satisfatório.

No entanto, este trabalho abre oportunidades para continuar o desenvolvimento do sistema e corrigir a limitação existente, a qual poderá ser resolvida com a utilização de antenas RFID de longo alcance. Outros tópicos de estudo são: a necessidade de se verificar os atrasos, latências e qualidade do sinal para a configuração proposta, o que pode afetar sua aplicação, pois a marcação correta de tempo é fundamental nesta situação.

REFERÊNCIAS

- ALL ABOUT ELETRONIC CIRCUITS. **Multi layer air core inductor calculator**. 2010. Disponível em: <http://www.circuits.dk/calculator_multi_layer_aircore.htm>. Acesso em: 06 de dez. 2019.
- ARDUINO. **Arduino UNO R3**. 2019. Disponível em < <https://store.arduino.cc/usa/arduino-uno-rev3>> arduino-uno-rev3>. Acesso em: 06 dez. 2019.
- ARDUINO. **Getting Started with Arduino UNO**. 2019. Disponível em: <<https://www.arduino.cc/en/Guide/ArduinoUno>>. Acesso em: 06 de dez. 2019.
- BRASIL. [Constituição (1988)]. **Constituição da República Federativa do Brasil: promulgada em 5 de outubro de 1988**. Senado Federal. Brasília. DF. 1988. Disponível em: <http://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm>. Acesso em: 06 de dez. 2019.
- CHANDLER, A.D. **O século eletrônico**. Editora Campus, Rio de Janeiro, RJ, 2002.
- GONSALES, S. **Etiquetas RFID revolucionando a gestão de estoques**. 2017. Disponível em :<<https://www.ecommercebrasil.com.br/artigos/etiquetas-rfid-revolucionando-gestao-estoques/>>. Acesso em 06 de dez. 2019
- GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO. **Modos de comunicação**. 2012. Disponível em: <https://www.gta.ufrj.br/grad/12_1/rfid/links/modos_de_comunicacao.html>. Acesso em: 06 de dez. 2019.
- LABORATÓRIO DE GARAGEM. **Tutorial: Comunicação SPI (Serial Peripheral Interface) com Arduino**. 2012. Disponível em: <<http://labdegaragem.com/profiles/blogs/tutorial-comunica-o-spi-serial-peripheral-interface-com-arduino>>. Acesso em 06 de dez. 2019.
- MARESPIERA. **Diferença entre protótipos alfa e beta**. 2019 Disponível em: <<http://marespiera.com/category/tecnologia/diferenca-entre-alfa-e-beta-prototipo.php>>. Acesso em: 06 de dez. 2019.
- MATINO, J.M. **EA078 Micro e Minicomputadores: Hardware**. 2004. Disponível em: < http://www.dca.fee.unicamp.br/courses/EA078/1s2004/arquivos/turma_ab/cap8.pdf> Acesso em: 06 de dez. 2019.
- MAXIM INTEGRATED PRODUCTS. **DS3231 Extremely Accurate I2C-Integrated RTC/TCXO/Crystal**. 2015
- MENDONÇA, H.S. **SPI E I2C**. 2016. Disponível em <<https://paginas.fe.up.pt/~hsm/docencia/comp/spi-e-i2c/>> Acesso em: 06 de dez. 2019.

MICROSOFT. **Bem-vindo ao IDE do Visual Studio.** 2019. Disponível em: <<https://docs.microsoft.com/pt-br/visualstudio/get-started/visual-studio-ide?view=vs-2019>>.

Acesso em: 06 de dez. 2019.

MICROSOFT. **Introdução à linguagem C# e ao .NET Framework.** 2015. Disponível em: <<https://docs.microsoft.com/pt-br/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>>.

Acesso em: 06 de dez. 2019.

MICROSOFT. **Introdução ao VBA no Office.** 2019. Disponível em: <<https://docs.microsoft.com/pt-br/office/vba/library-reference/concepts/getting-started-with-vba-in-office>>.

Acesso em: 06 de dez. 2019.

MORAES, C.; CASTRUCCI, P. **Engenharia de Automação Industrial.** 2. ed. Rio de Janeiro. LTC, 2007.

MUNDO ELETRÔNICA. **Cartões RFID e o Módulo MFRC522.** 2019. Disponível em: <<https://www.moduloeletronica.com.br/blog?single=Cartoes-RFID-e-o-Modulo-MFRC522>>.

Acesso em 06 de dez. 2019

NX SEMIC. **Standart performance MIFARE and NTAG fronted.** 2016

ORTEGA, J. **Indústria 4.0: entenda o que é a quarta revolução industrial.** 2019, Disponível em <<https://www.startse.com/noticia/nova-economia/60414/industria-4-0-entenda-o-que-e-quarta-revolucao-industrial>>, Acesso em 06 dez. 2019

SILEVIRA, L.; LIMA, W.Q. **Um breve histórico conceitual da Automação Industrial e Redes para Automação Industrial.** 2003. Artigo – Universidade Federal do Rio Grande do Norte, Natal, RN, 2003.

WHEELER, H. A. Simple Inductance Formulas for Radio Coils. **Proceedings of the I.R.E.**, v. 16, n. 75, p. 1398-1400. Out., 1928.

ZILLER, R.M.; **“Microprocessadores – Conceitos Importantes,”** Edição do autor, Florianópolis, SC, 2000. ISBN 85-901037-2-2.

APÊNDICES

APÊNDICE A – Código utilizados

Os códigos utilizados estão disponíveis no site GitHub que pode ser acessado pelo link
<<https://github.com/leticiamorinaga/TCC>>