CENTRO UNIVERSITÁRIO UNIFACVEST
CURSO DE CIÊNCIA DA COMPUTAÇÃO
ELIANE NUNES COSTA

ACEEJ: SISTEMA EMBARCADO DE PROGRAMAÇÃO DE SINAL DE INTERVALO ESCOLAR

Lages - SC
2016

ELIANE NUNES COSTA

ACEEJ: SISTEMA EMBARCADO DE PROGRAMAÇÃO DE SINAL DE INTERVALO ESCOLAR

Trabalho de conclusão de curso apresentado ao Centro Universitário UNIFACVEST como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação.

Prof. Me. Márcio José Sembay

Co-orientador:
Prof. João Francisco Frank Gil
Prof. Igor Muzeka

Lages - SC
2016

ELIANE NUNES COSTA

ACEEJ: SISTEMA EMBARCADO DE PROGRAMAÇÃO DE SINAL DE INTERVALO ESCOLAR

Trabalho de conclusão de curso apresentado ao Centro Universitário UNIFACVEST como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação.

Prof. Me. Márcio José Sembay

Co-orientador:
Prof. João Francisco Frank Gil
Prof. Igor Muzeka

Lages, SC____/____/2016. Nota____ _____

_____
Prof. Me. Márcio José Sembay

Lages - SC
2016

AGRADECIMENTOS

Primeiramente agradeço a Deus por estar sempre me guiando os meus passos na caminhada da vida. Agradeço a todos os professores que me apoiaram e me orientaram, mas em especial ao professor e coordenador do curso Marcio Sembay, companheiro de caminhada ao longo do Curso de Ciência da Computação. Eu posso dizer que a minha formação, inclusive pessoal, não teria sido a mesma sem a sua pessoa. Agradeço também ao meu esposo, Joacir Hugens de Liz, que de forma especial, com carinho e paciência me deu força e coragem, me apoiando nos momentos de dificuldades. Aos meus amigos, que embora não tivessem conhecimento disto, mas me iluminaram de maneira especial os meus pensamentos me levando a buscar mais conhecimento. E não deixando de agradecer de forma grata e grandiosa aos meus pais, Cosme Salviano Costa e Antônia Nunes da Costa (*in memorian*), a quem eu agradeço todas as noites a minha existência. À minha irmã Elisangela Nunes Costa, que mesmo longe vibra a cada vitória minha e ao meu sobrinho ao qual tenho tanto carinho.

ACEEJ: SISTEMA EMBARCADO DE PROGRAMAÇÃO DE SINAL DE INTERVALO ESCOLAR

[1]Eliane Nunes Costa
[2]Márcio José Sembay
[3]João Francisco Frank Gil
[4]Igor Muzeka

RESUMO

Com os grandes avanços na área da tecnologia pode-se observar que as aplicações trazem inúmeros benefícios no cotidiano escolar. Estes avanços viabilizam as construções de ambientes inteligentes e interativos onde a computação é utilizada para melhorar cada vez mais atividades comuns do dia a dia. Estas aplicações são munidas de sensores, microcontroladores capazes de interagir com outros *softwares* ou serem independentes, facilitando e otimizando as operações a serem realizadas. Neste projeto foi desenvolvido o Aceej com o objetivo de criar uma ferramenta que viabilizasse o trabalho manual dos profissionais da escola, alterando a tarefa manual de acionar o sinal de intervalo, para uma forma automática e configurável de modo simples e eficaz. Sendo proposto o levantamento das características e operação dos sistemas embarcados com Arduino, visando seu uso em sistemas de pequeno porte. Foi realizado um estudo tecnológico de caráter bibliográfico, em que foi possível compreender as situações para o desenvolvimento deste projeto. O desenvolvimento do Aceej é de um sistema embarcado capaz de acionar e controlar o acionamento do sinal de intervalo escolar. Neste trabalho foi utilizada a plataforma Arduino, que foi desenvolvido em uma interface gráfica em C# que permite ao usuário alterar os horários de acionamento do sinal. O principal objetivo deste trabalho é possibilitar o controle e automatização do sinal, proporcionando mais praticidade ao usuário.

Palavras Chave: Ambientes Inteligentes, Sistema Embarcado, Automatização, Computação.

---

[1] Acadêmica do Curso de Ciência da Computação, 7ª fase do Centro Universitário UNIFACVEST
[2] Mestre em Ciência da Informação pela UFSC
[3] Especialista em Redes de computadores, Mestrando em práticas transculturais
[4] Graduado em Ciência da Computação, Pós em Engenharia de Sistemas e a Ciência da Computação e Mestrando em práticas transculturais.

ACEEJ: SIGNAL PROGRAMMING BOARDED SYSTEM SCHOOL BREAK

[5]Eliane Nunes Costa
[6]Márcio José Sembay
[7]João Francisco Frank Gil
[8]Igor Muzeka

ABSTRACT

With the great advances in the area of technology it can be observed that the applications bring numerous benefits in the school routine. These advances enable the construction of intelligent and interactive environments where computing is used to improve more and more everyday activities. These applications are equipped with sensors, microcontrollers capable of interacting with other software or being independent, facilitating and optimizing the operations to be performed. In this project Aceej was developed with the objective of creating a tool that would enable the manual work of the professionals of the school, changing the manual task of activating the interval signal, for an automatic and configurable form in a simple and effective way. It is proposed the survey of the characteristics and operation of the systems shipped with Arduino, aiming its use in small systems. A technological study of a bibliographic character was carried out, in which it was possible to understand the situations for the development of this project. The development of Aceej is an embedded system capable of activating and controlling the activation of the school interval signal. In this work we used the Arduino platform, which was developed in a graphical C # interface that allows the user to change the signal activation times. The main objective of this work is to enable the control and automation of the signal, providing more convenience to the user.

Keywords: Intelligent Environments, Embedded System, Automation, Computing.

---

[5] Acadêmica do Curso de Ciência da Computação, 7ª fase do Centro Universitário UNIFACVEST.
[6] Mestre em Ciência da Informação pela UFSC
[7] Especialista em Redes de computadores, Mestrando em práticas transculturais
[8] Graduado em Ciência da Computação, Pós em Engenharia de Sistemas e a Ciência da Computação e Mestrando em práticas transculturais.

ACCEJ: SEÑAL DE PROGRAMACIÓN DE INTERRUPCIÓN DEL SISTEMA ABORDAMOS ESCUELA

[9]Eliane Nunes Costa
[10]Márcio José Sembay
[11]João Francisco Frank Gil
[12]Igor Muzeka

RESUMEN

Con los grandes avances en la tecnología puede verse que las aplicaciones traer muchos beneficios en la vida escolar cotidiana. Estos avances permiten la construcción de entornos inteligentes e interactivos donde se utiliza la informática para mejorar cada vez más comunes del día a día. Estas aplicaciones están provistos de sensores, microcontroladores capaces de interactuar con otro software o ser independiente, facilitando y optimizando las operaciones a realizar. Este proyecto fue desarrollado Aceej con el fin de crear una herramienta que haga posible el trabajo manual de profesionales de la escuela, el cambio de la tarea manual de conducir la señal de intervalo de una manera sencilla y eficaz, automático y configurable. Se propone el levantamiento de las características y el funcionamiento del sistema integrado con Arduino, con el objetivo de su uso en pequeños sistemas. Un estudio tecnológico de referencias bibliográficas, que era posible comprender las situaciones para el desarrollo de este proyecto se llevó a cabo. El desarrollo de un Aceej incorporado es capaz de dirigir y controlar la unidad de alarma de rango escuela. En este trabajo se utilizó la plataforma Arduino, el cual fue desarrollado en una interfaz gráfica en C# que permite al usuario cambiar los tiempos de conducción de señales. El principal objetivo de este trabajo es permitir el control y la automatización de la señal, proporcionando una mayor comodidad para el usuario.

Palabras Clave: Entornos Inteligentes, Sistemas de Respaldo, Automatización, Informática.

---

[9] Acadêmica do Curso de Ciência da Computação, 7ª fase do Centro Universitário UNIFACVEST.
[10] Mestre em Ciência da Informação pela UFSC
[11] Especialista em Redes de computadores, Mestrando em práticas transculturais
[12] Graduado em Ciência da Computação, Pós em Engenharia de Sistemas e a Ciência da Computação e Mestrando em práticas transculturais.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE SIGLAS

COLL	-	*C-Like Object Oriented Language*
IDE	-	Ambiente de Desenvolvimento Integrado
LCD	-	*Liquid Crystal Display*
PC’s	-	*Personal Computers*
PWM	-	*Pulse Width Modulation*
SO’s	-	Sistemas Operacionais

SUMÁRIO

1 INTRODUÇÃO

A era da transformação digital já está batendo na porta de todas as pessoas de uma forma incontestável. As mudanças relacionadas ao uso das tecnologias nos mais variados sentidos da sociedade e dos negócios, vem proporcionando novos tipos de inovação que vão muito além de aprimorar simples transações da rotina diária e métodos na rotina de trabalho. Muitos cientistas e pensadores respeitados acreditam que no futuro, em todos os momentos da vida será equipada com a evolução dos atuais equipamentos eletroeletrônicos e dispositivos altamente inteligentes. Esse futuro foi descrito no artigo "*This new House*" por Murphy (2005), da revista *Fortune*, e pode se observar que não parece tão impossível de acontecer, comparando com a evolução eletrônica que está trazendo cada vez mais comodidade às pessoas.

Os sistemas embarcados estão presentes em diversas áreas, desde a automação residencial, automotiva, empresas e indústrias. É praticamente impossível imaginar as tarefas do dia a dia sem os sistemas eletrônicos. Conforme uma pesquisa realizada pela Universidade do Paraná em 2014, evidencia-se que uma pessoa exerce interação diária com pelo menos 20 sistemas embarcados e que em 2020, uma pessoa terá aproximadamente um convívio com 350 processadores por dia (SUEIRO, 2014).

Os sistemas embarcados são constituídos de *hardware* e *softwares* e executam funções específicas, possuindo recursos limitados e requisitos de tempo real. Grande parte dos sistemas embarcados são desenvolvidos utilizando microcontroladores pequenos e com preço baixo (BANNATYNE, 2009).

Esses sistemas possuem muitas vantagens, pois facilita o gerenciamento de multitarefas, de tempo e de recursos. Tudo que é criado para otimizar tempo e dinheiro é bem visto tanto ao cliente quanto para a empresa. Com toda essa evolução na tecnologia atualmente, não há limites para as criações de projetos, sejam eles grandes ou pequenos, pois a junção das criações já existentes com inovações pode solucionar problemas do cotidiano que ainda não foram resolvidos. Tendo como base o conhecimento de que os sistemas embarcados estão em praticamente todas as tarefas do dia a dia, origina-se o questionamento que motiva a criação deste projeto: De que forma um sistema embarcado em plataforma Arduino pode auxiliar no gerenciamento e controle dos sinais de intervalos escolares?

1.1 Justificativa

Este é um projeto sobre o desenvolvimento de um sistema embarcado que programará o sinal de intervalo escolar. Foi escolhido este tema devido ao coordenador do curso de Ciência da Computação Marcio Sembay e meu esposo Joacir Hugens de Liz trabalhar em um ambiente escolar e ter sido identificado este problema no dia a dia nas escolas. Os sistemas embarcados hoje estão envolvidos em muitas atividades cotidianas. Então foi associado a necessidade escolar a uma ferramenta que já é bastante utilizada. Na elaboração deste projeto propõe-se a criar um sistema embarcado capaz de efetuar e tratar os sinais de intervalo da escola, com diferentes opções de forma eficiente, não será necessário um *hardware* muito robusto e que seja de difícil utilização. Serão utilizadas plataformas *open source*, que desta maneira é possível reduzir o custo do projeto.

1.2 Importância

Para auxiliar no desenvolvimento e principalmente trazer melhorias na utilização, o dispositivo funciona em um sistema operacional embarcado. Desta forma alguns recursos de *hardware* e *firmware* serão administrados pelo sistema operacional.

A inserção de novas informações pelo usuário é de fácil utilização e não causará riscos de alterar alguma função já preestabelecida. No desenvolvimento deste trabalho foi utilizado um *hardware* propagado no mercado e que é de fácil aquisição com preço bem baixo.

A aplicação desenvolvida contém algumas funcionalidades, sem alterar o vínculo com o *hardware*, bastando apenas colocar as informações específicas que automaticamente são direcionadas às configurações gerais do dispositivo.

1.3 Objetivos Gerais

Criar uma ferramenta que viabiliza o trabalho manual dos profissionais da escola, alterando a tarefa manual de acionar o sinal de intervalo para uma forma automática e configurável de modo simples e eficaz.

1.3.1 Objetivos Específicos

Reconhecer as necessidades estabelecidas e realizadas em determinadas atividades, extinguindo ou pelo menos as minimizando trazendo um sucesso considerável a um projeto. O processo de automatizar e facilitar a vida do profissional escolar será o principal objetivo deste trabalho.

Serão listadas abaixo as funcionalidades imprescindíveis a serem desenvolvidas para este projeto:

a) Temporizar os acionamentos de forma precisa e previsível o tempo exato que deverão ocorrer os acionamentos da sirene;

b) Controlar em tempo a execução repetidamente dos horários definido pelo usuário com um intervalo de tempo específico entre cada acionamento;

c) Gerenciar mudanças de horários conforme a gestão escolar venha necessitar, permitindo a melhor gestão do *software*.

2 REVISÃO LITERÁRIA

Neste trabalho, foi realizado uma busca pelos conhecimentos existentes, abordando visões de vários autores, e explanando os conhecimentos adquiridos sobre Sistemas Embarcados, Sistemas Operacionais Embarcados, Linguagem C# e da plataforma Arduino onde será desenvolvido o projeto.

2.1 Breve Histórico

Nos anos 40, os computadores eram específicos para realizar uma única tarefa e não eram comerciais, eram muito grandes para serem considerados embarcados e foram utilizados apenas para fins militares pelo governo americano.

O primeiro Sistema Embarcado mundialmente reconhecido no início dos anos 60 foi o Apollo Guidance Computer, desenvolvido nos EUA por Charles Stark Draper no Laboratório de Instrumentação MIT para a NASA. O primeiro Sistema Eletrônico embarcado que se tornou uma produção em grande escala foi o computador guia do míssil nuclear norte-americano LGM-30 Míssil Minuteman, lançado em 1961.

Em 1966 foi lançada uma nova versão do Minuteman que era constituído por circuitos integrados. Esta produção em grande escala tornou a tecnologia disponível para comercializar. Desde aquela época os Sistemas Embarcados têm reduzido consideravelmente de tamanho e cada vez mais aumentando sua eficácia, mas mantendo um preço comercial (MAGNUN, 2015).

Figura 1: Um dos primeiros Minuteman

Fonte: https://commons.wikimedia.org/wiki/File:Minuteman_I.jpg

Diferentemente de um PC que pode executar muitos programas e intercalar entre eles, exercendo várias funções, os sistemas embarcados são dispositivos invisíveis, que fazem parte do nosso cotidiano, muitas pessoas não possuem conhecimento que usem diariamente, mas eles fazem parte da nossa rotina diária.

São produzidos basicamente, pelos mesmos elementos de um PC sendo eles: memória, dispositivo de armazenamento, processador, interfaces, entre outros, que segue a configuração de um *chip* para cada função ou todas as funções agrupadas em um único *chip*, agrupando todos os circuitos. A principal disparidade de um PC ou de um smartphone é que os sistemas embarcados são especializados em executar uma única tarefa específica. Basicamente, qualquer equipamento independente que não seja um PC ou qualquer outro tipo de computador pessoal, se enquadra em um sistema embarcado (DELAI, 2013).

2.1.1 Sistemas Embarcados

Sistemas Embarcados é um sistema eletrônico que tem como princípio um microprocessador, que difere de um computador pessoal, possui um *software* completamente específico ao dispositivo ou sistema que ele controla. Assim, ao contrário dos computadores de uso pessoal, um sistema embarcado realiza um conjunto de tarefas pré-definidas, geralmente com requisitos específicos. O sistema é dedicado a tarefas específicas, com redução de tamanho e de recursos computacionais, assim o custo do produto final não se torna tão alto (EMBARCADOS, 2013).

2.1.2 Sistemas Operacionais embarcados

Um PC de uso geral ou de uso específico é utilizado para armazenamento ou manipulação de dados, tudo o que ele realiza é instruído a partir de *softwares*. Para que seja possível este armazenamento, manipulação, redirecionamento ou modificação de dados é necessário o uso de periféricos que enviam ou recebem informações de forma lógica (MORIMOTO, 2007).

Os Sistemas Embarcados geralmente utilizam poucos recursos de memória, processamento, portanto não são idealizados para utilizar SO’s em PC de uso pessoal, nos Sistemas Embarcados, compreende-se como qualidade a segurança de que o *software* não apresente falhas (HAMACHER et al., 2012; AKYILDIZ et al., 2002).

2.2 Definição de sistemas embarcados

Sistemas embarcados é qualquer dispositivo eletrônico ou qualquer sistema com eficácia de processar e armazenar informações, executando uma função de forma que se repete sem falha de sistema. Um sistema embarcado deve ser íntegro e livre para executar uma determinada função (CUNHA, 2007; CHASE, 2007). Compreende-se uma ligação de componentes mecânicos, *Software* e *Hardware*, desenvolvidos para a aplicação de uma função específica (LI, 2003; STALLINGS, 2012).

Um sistema embarcado é um sistema bastante limitado, com restrições de memória, tamanho, energia, potência, possuindo uma particularidade, de funções. Na maior parte está

agregado em outro produto. Possuindo em muitas vezes características de sistemas de tempo real, com alta velocidade de envio e recebimento de informações (TANENBAUM, 2009).

Os Sistemas Embarcados estão em muitos aparelhos utilizados no dia a dia de cada pessoa, espalhados em diversas áreas, como eletrônicos em geral, controle industrial, instrumentos médicos, dispositivos de rede, indústria automobilística (NOERGAARD, 2005).

No quadro abaixo é possível visualizar diferentes exemplos de Sistemas Embarcados no mercado e que eles estão presentes em praticamente todas as atividades cotidianas, e com as variedades e baixos custos atuais, tende a aumentar o uso no cotidiano das pessoas.

Quadro 1: Exemplos de Sistemas Embarcados no mercado.

Exemplos de Sistemas Embarcados	
Mercado	Dispositivo Embarcado
Automotivo	Sistema de Ignição
	Controle de Motor
	Freio (ABS)
Eletrônica de consumo	Televisores Analógicos e Digitais
	DVD's, Video Cassetes
	Personal Data Assistants (PDAs)
	Eletrodomésticos (Refrigeradores, Microondas, Torradeiras)
	Brinquedos, Jogos
	Telefones, *Pagers* e Celulares
	Câmeras
	Global Positioning Systems (GPS)
Controle Industrial	Robótica e Controle de Sistemas (Manufatura)
Medicina	Bombas de Infusão
	Próteses
	Equipamento de Diálise
	Monitores Cardíacos
Redes de Comunicação	Roteadores
	Hubs
	Gateways
Automação de Escritórios	Equipamento de Fax
	Copiadoras
	Impressoras
	Monitores Cardíacos
	Scanners

Fonte: NOERGAARD, 2005.

O aumento de sistemas embarcados, torna o mercado bem mais atraente para novos fornecedores e empresas solidas, deixando o mercado largamente aberto, diferente do mercado de SO's para PC's, (ORTIZ, 2001; LI, 2003; NOERGAARD, 2005; STALLINGS, 2012).

Os Sistemas embarcados têm despertado fascínio em vários segmentos de mercado nos últimos anos, a evolução das tecnologias neste tipo de sistemas tem sido muito grande e sua definição é dada a uma vasta variedade de tecnologias empregadas.

2.2.1 Benefícios do Sistema Embarcado

A grande parte dos sistemas embarcados existentes atualmente são desenvolvidas utilizando microcontroladores, com preço baixo e de pequeno porte (BANNATYNE, 2009). A maior parte dos sistemas embarcados é segmentada por benefícios importantes (BERGER, 2002; BUTTAZO, 2006):

a) Processadores diferentes: sistemas embarcados são suportados por arquiteturas de processadores e grande quantidade de processadores.

b) Controle de tempo real: os dispositivos embarcados interagem com o ambiente e responde aos comandos executados dentro do tempo estabelecido, garantindo os requisitos de desempenho *off-line.*

c) Desempenho eficiente: executa dezenas ou centenas de funções que se comunicam entre si para o uso de recursos compartilhados.

d) Custos frágeis: devido à grande produção pelas indústrias altamente competitivas os custos se tornam reduzidos. Portanto, as aplicações embarcadas trabalham com poucas unidades de processamento, memória e pouco potencial computacional, apesar de que para obter retorno dos custos é obrigatório o uso de recursos computacionais eficientes.

A principal atividade de um *software* embarcado é a comunicação com o ambiente físico e não de transformação de dados (LEE, 2002).

2.2.2 Requisitos funcionais e não funcionais de um Sistema Embarcado

Os requisitos de um projeto podem ser funcionais e não funcionais eles são de suma importância para o sucesso do projeto. Os requisitos funcionais expõem as funcionalidades que se espera de um projeto, ou seja, que ele funcione de forma precisa e consistente.

Os requisitos não funcionais refletem propriedades do sistema, como a performance do sistema, o custo de produção, memória, o tamanho, o peso e o consumo de energia. Está diretamente correlacionada à aplicabilidade de um sistema, também se refere aos atributos de qualidade.

O quanto mais cedo se define os requisitos não funcionais tão logo se evitarão inconsistências e mau funcionamento. Realizando a análise dos requisitos e articulando sua arquitetura irá conservar sua subsistência e facilitar a adição de novas funcionalidades (FILHO SILVA, 2008).

2.2.3 Desenvolvimento do Sistema Embarcado

Sistemas Embarcados representam um ambiente e são desenvolvidos para tarefas específicas, isto faz com que eles possam ser altamente independentes, que por sua vez as limitações a serem executadas são bem definidas (BERGER, 2002).

Grande parte dos Sistemas Embarcados faz uso de microcontroladores, pois os mesmos contem periféricos e memória integrados num *chip*, em contraste com os microprocessadores que são utilizados em computadores pessoais, ou outras aplicações (BERGER, 2002; NOERGAARD, 2005; BRAGA, 2001).

No desenvolvimento de Sistemas Embarcados propõe-se um comportamento abrangente que integre de forma sólida, princípios de teoria, desenvolvimento de *hardware* e de *software* (SIFAKIS; HENZINGER, 2006).

Existem diferentes formas no ciclo do desenvolvimento dos sistemas embarcados, que são baseadas nas listadas abaixo (NOERGAARD, 2005):

a) Modelo Espiral: no processo de desenvolvimento há várias etapas e no final, são realizados *feedbacks* no decorrer do processo.

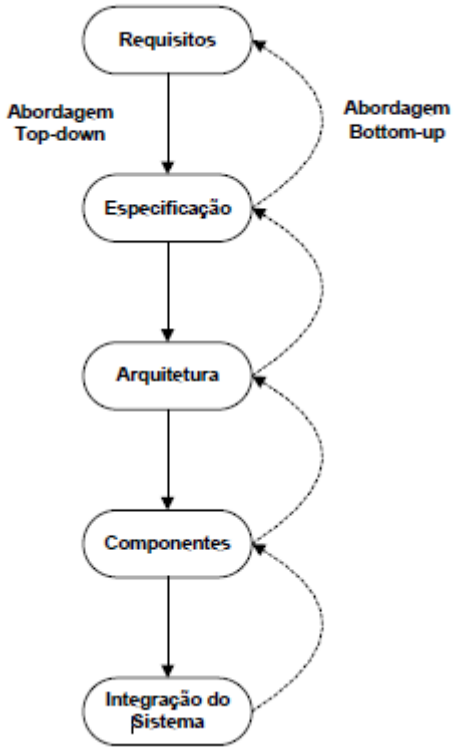
b) Modelo *big-bang*: não há uma regra ou estilo no desenvolvimento do sistema.

c) Modelo *code-and-fix:* são desenvolvidos os requisitos do produto, mas não há nenhuma formalidade antes de iniciar o desenvolvimento.
d) Modelo cascata: é desenvolvido em etapas, onde se torna requisito da próxima a conclusão da etapa anterior.

No desenvolvimento de um sistema embarcado na abordagem *top-down* se define em: definição dos requisitos, especificação, arquitetura, componentes e a integração do sistema. Começando o desenvolvimento com os requisitos, descrição detalhada com precisão nos detalhes, para qual arquitetura será desenvolvida e a estrutura dos componentes ao sistema, assim será construído (WOLF, 2008).

Na imagem abaixo é possível observar a abordagem *top-down*, no desenvolvimento de sistemas embarcados. Nas linhas continuas aborda-se *Top-down* e as linhas tracejadas a *Bottom-up* (WOLF, 2008).

Figura 2: Principais etapas no desenvolvimento de Sistemas Embarcados.



Fonte: WOLF, 2008.

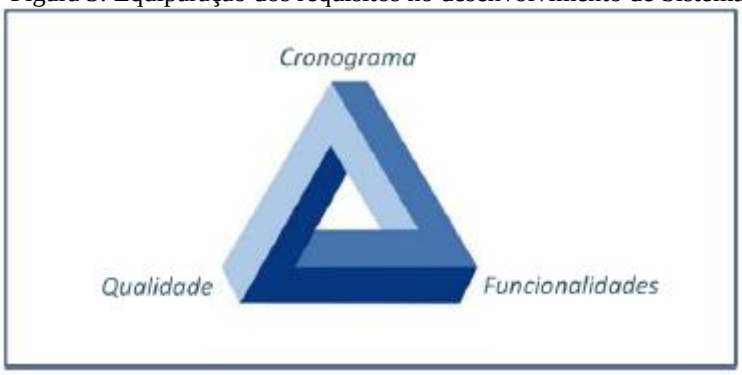
Berger (2002) descreve os mesmos princípios em seu livro, salientando a realização dos testes, iteração e refinamento, manutenção e atualizações. As etapas descritas por ele são:

a) Especificação do produto
b) Divisão de desenvolvimento em componentes de *hardware* e *software*
c) Iteração e refinamento da divisão
d) Desenvolvimento independente de *hardware* e *software*
e) Teste e lançamento do produto
f) Atualizações e manutenções

A utilização de uma metodologia de desenvolvimento permite a automatização de etapas, garantindo a execução de todas as atividades (WOLF, 2005).

Um desequilíbrio nos requisitos pode colocar o desenvolvimento de um produto em risco, quando se está desenvolvendo um produto deve se equiparar as premissas do cronograma, a qualidade e as funcionalidades. Na (figura 3) pode se observar esses três princípios (GANSSLE, 2008).

Figura 3: Equiparação dos requisitos no desenvolvimento de Sistemas Embarcados



Fonte: GANSLLE, 2008.

3 PLATAFORMA ARDUINO

O Arduino é uma plataforma *Open Source*, baseada em *hardware* e *software* livre altamente flexível e fácil de usar (CULKIN, 2012).

O Arduino é capaz de comunicar-se com o ambiente pelo resultado de uma multiplicidade de sensores que emitem sinais de entrada e interpreta por meio de sinais de saídas executando a função predeterminada. O microcontrolador na placa do Arduino utiliza a programação do Arduino que é fundamentado no framework wiring que é uma ferramenta *Open Source*, que permite desenvolver *softwares* multiplataforma (BARRAGÁN, 2012).

As placas podem ser adquiridas com facilidade por um preço baixo, e o *software* pode ser baixado gratuitamente. Tornando-se assim um *software* de referência por ser possível adapta-lo às necessidades. Os projetos realizados com Arduino podem se interagir com outros *softwares* ou podem ser independentes (ARDUINO, 2016)

3.1 Arquitetura de Hardware

O Arduino Uno foi fabricado em 2005, na Itália por David Cuartielles e Massimo Banzi, com o objetivo de produzir uma ferramenta que não tivesse complexidade e com baixo custo (SILVEIRA, 2011).

Sua plataforma é constituída com 14 pinos de entrada/saída sendo 6 deles com saída PWM, sendo ideal para projetos menores devido a ter menor quantidade de pinos. Neste projeto será utilizado o Arduino Uno, devido a não ser um projeto de grande complexidade e não haver a necessidade de utilização de muitas portas.

A plataforma do Arduino Mega é constituída por um microcontrolador da marca *Atmel*, com 54 pinos de entradas e saídas, entre elas analógicas e uma interface USB responsável pela comunicação. A placa possui tamanho consideravelmente pequeno para a quantidade de pinos que possui (SOUZA, 2014). Sua utilização é dada em projetos maiores e mais complexos.

O Arduino Uno é uma plataforma de prototipagem eletrônica, projetada com um microcontrolador Atmel, com suporte de entrada/saída embutidas e um ambiente de desenvolvimento (IDE) que implementa e atua com linguagens (C/C++). É possível remodelar o estado de um determinado ambiente, controlando luzes, abrindo portas ou portões eletrônicos por exemplo. Na sua interface é concebível observar, com base em uma diversidade de sensores,

digitais e/ou analógicos, o que pode acontecer em alguns ambientes (MCROBERTS, 2011). Na aplicação proposta será utilizado o Arduino Uno.

3.2 Simulação do Hardware para Arduino

Existem vários programas para simulação de *hardware* para o Arduino, será apresentado o 123D Circuits, da Autodesk que é simples e fácil de explorar. A Autodesk dispõe de ferramentas de simulação 3D há anos. O 123D Circuits é *open source* e funciona direto no navegador, não sendo necessário instalar nada. Além de simular circuitos eletrônicos que incluem o Arduino, o 123D Circuits permite que seja criado o desenho de uma placa de circuito impresso com o circuito que foi projetado e simulado. Sendo possível também ter uma visão do proto-board, a simulação em funcionamento do programa do Arduino e da parte eletrônica, entre outras. Para acesso basta entrar no link https://123d.circuits.io/lab e explorar o simulador em tempo real, podendo ser tanto iniciante como alguém que já possua bastante experiência. Neste site também é possível visualizar através de um repositório, trabalhos já realizados por outras pessoas (AUTODESK, 2016).

3.3 Software Arduino e Linguagem C

Arduino é uma plataforma *Open Source*, que além de ser um conjunto de ferramentas, possibilita a criação de aparelhos eletrônicos. O Arduino é tanto um *hardware* quanto um *software*, sua idealização é possibilitar que desenvolvedores criem aplicativos específicos para funcionar em circuitos eletrônicos. O Arduino pode funcionar de maneira independente, sem que seja necessário agregar outro *software* mais complexo para realizar seu funcionamento, na placa do Arduino ele pode receber vários sinais, devidos aos seus sensores, e ele se utiliza disso para se comunicar também com outros aparelhos ou até mesmo se comunicar com aplicativos existentes em um computador.

O Arduino é extremamente flexível, pois dispõe ao desenvolvedor a confiança e liberdade de criar algo mais simples ou montar algo grandioso. O Arduino é uma plataforma de prototipagem eletrônica fundamentado na linguagem Wiring que usa uma interface gráfica construída em Java baseada no ambiente *Processing*. Consiste em um programa IDE muito simples de usar e de compreender com bibliotecas que são facilmente encontradas na internet.

A IDE do Arduino possui uma linguagem própria baseada na linguagem C e C++. A sintaxe da linguagem de desenvolvimento para o Arduino é baseada na linguagem de programação de alto nível C. Oferecendo praticamente as mesmas estruturas básicas da linguagem C: condicional usando *if, then e else; switch e case*; e estruturas de repetição *while*, *do while* e *for*; funções e procedimentos; *structs*; etc.

O *software* é *Open Source* o que torna mais acessível a todas as pessoas que desejam iniciar seus projetos, sendo possível baixar o *software* do Arduino através do site: https://www.arduino.cc/en/Main/Software.

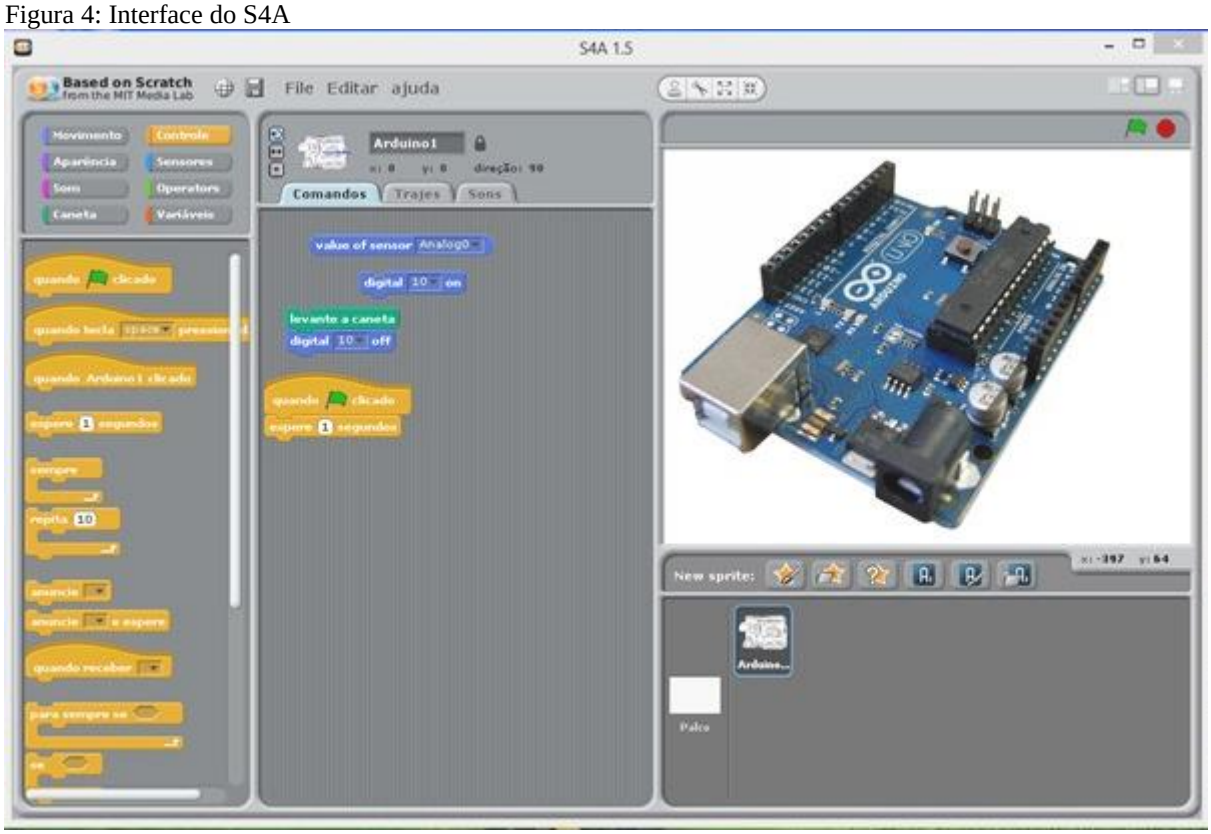
3.4 Linguagens de Programação para Arduino

Existem linguagens alternativas para criar na plataforma do Arduino, como por exemplo o S4A que é uma modificação do Scratch[13], foi desenvolvido no ano de 2010 pela equipe Smalltalk do Citilab. O S4A é um ambiente de desenvolvimento simples que facilita o aprendizado na plataforma de *hardware* do Arduino. Seu ambiente é através de blocos de comandos com funcionalidades básicas onde é possível ler e enviar valores de todos os pinos.

Sua interface é fácil e simples de utilizar, com objetivo de cada vez mais atrair pessoas para o mundo da programação. O S4A funciona com o Arduino Diecimila, Duemilanove e Uno, outras placas não foram testadas, mas não impede que não possam ser utilizadas. O S4A pode ser baixado através do site: http://s4a.cat/.

[13] Scracth é um projeto do *Lifelong Kindergarten Group do MIT Media Lab*. Disponibilizado gratuitamente. Utilizado para jogos e animações interativos, ajudando jovens a raciocinar e criar, podendo partilhar suas criações em uma comunidade.

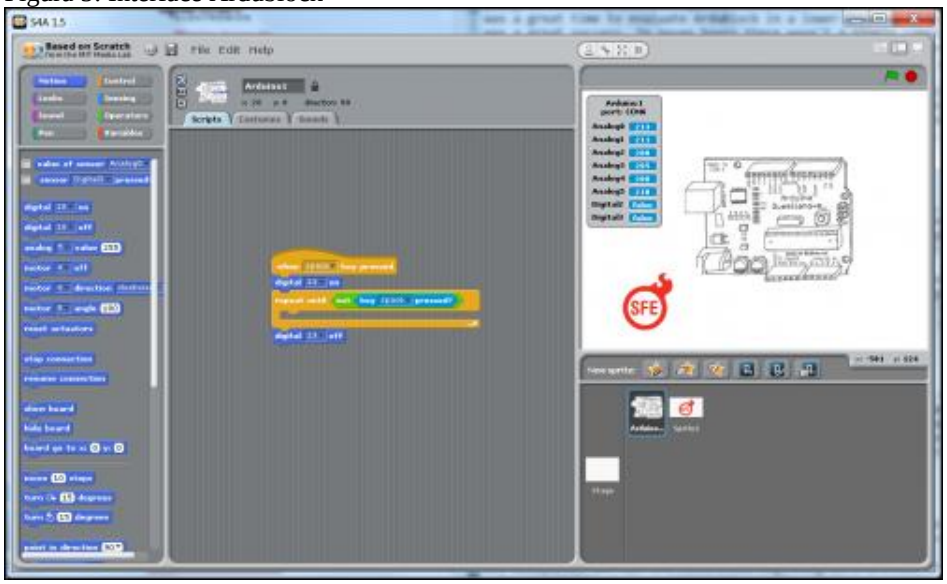
Figura 4: Interface do S4A



Fonte: http://pplware.sapo.pt/gadgets/hardware/s4a-programar-o-arduino-e-facil-e-divertido/

A plataforma Ardublock também foi criada similar com o Scratch, que podem ser criados programas para Arduino por meio de blocos já estabelecidos, o que auxilia quem deseja entrar no mundo da programação e da eletrônica, também com uma interface bem fácil (SOUZA, 2014).

Figura 5: Interface Ardublock



Fonte: http://www.portalmcu.com.br/2013/12/ardublock-programacao-em-blocos-para-arduino.html

4 PROGRAMAÇÃO EMBARCADA EM LINGUAGEM C#

4.1 Breve Histórico

O C# foi desenvolvido no ano de 1999, por vários desenvolvedores, mas foi atribuída principalmente a Anders Hejlsberg na Microsoft, com o objetivo de criar uma linguagem nova que de início foi chamada de COOL. Com a criação da plataforma .Net na metade de 2000, a Microsoft resolveu dar um novo nome à linguagem para C# (EMBARCADOS, 2015).

4.2 Linguagem C#

A linguagem C# surgiu em conjunto com a arquitetura .Net, consequentemente foi especialmente desenvolvida para esta plataforma. O C# foi inspirado por diversas outras linguagens, como por exemplo, o Delphi, C++ e Java. Com isso o C# passou a ser uma linguagem poderosa, tendo em vista que foram utilizados os melhores recursos de cada uma das linguagens base. Por ser uma linguagem que segue os padrões de orientação a objetos e desenvolvida exclusivamente para a plataforma .Net, se torna uma linguagem extremamente expressiva, mas também simples e fácil de aprender (EMBARCADOS, 2015).

O que facilita a compreensão em ligação à plataforma do Arduino devido a sintaxe do C# ser instantaneamente reconhecida pela familiarização com a linguagem C, C++ e também com a capacidade de ser escrita em aplicações embarcadas.

4.2.1 Desenvolvimento do Sistema Embarcado com C# e Arduino

Neste projeto foi desenvolvido uma aplicação embarcada para Windows na plataforma .Net utilizando o ambiente Visual Studio, com a linguagem C# de forma simples, com fáceis elementos visuais. A comunicação entre o computador e placa será realizada através de uma porta serial emulada por intermédio do driver da USB (SOUZA, 2014).

5 FERRAMENTAS DO PROJETO

No seguimento deste trabalho, serão detalhadas as principais ferramentas a serem utilizadas no desenvolvimento do projeto.

5.1 Arduino Uno

O Arduino Uno 2560 é uma placa de microcontrolador baseado no ATmega328P. Possuindo 14 pinos digitais de entrada / saída, dos quais 6 podem ser usados como saídas PWM, 6 entradas analógicas, um cristal oscilador de 16 MHz, uma conexão USB, um cabeçalho ICSP, e um botão de reset. Em sua alimentação é possível conectá-lo diretamente a um computador através de um cabo USB ou ligá-lo com um adaptador AC-CC de parede ou bateria para inicializar. O Arduino possui muitos recursos para se intercomunicar com o computador e com outros modelos de Arduino e até mesmo com outros microcontroladores (ARDUINO, 2016).

Figura 6: Demonstração da plataforma do Arduino Uno

Fonte: https://multilogica-shop.com/arduino-uno-r3

Conforme quadro abaixo é possível ver as especificações do Arduino Uno:

Quadro 2: Especificações técnicas do Arduino Uno

Microcontrolador	ATmega328
Tensão de Alimentação	5V
Tensão de Entrada (recomendado)	7-12V
Tensão de Entrada (limites)	6-20V
Portas digitais I/O	14 (das quais 6 podem ser usadas como saída PWM)
Portas analógicas de entrada	6
Corrente DC por pino	40 Ma
Corrente DC por pino 3.3V	50 Ma
Memória Flash	32KB of which 0,5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock	16 MHz

Fonte: https://www.arduino.cc/en/Main/arduinoBoardUno

5.1.1 Conversor de nível lógico 3,3 – 5V

É um conversor que possui 2 canais bidirecionais que convertem tensão de 3,3 para 5v e de 5v para 3,3v. Este conversor foi utilizado para efetuar a conversão do modulo do relógio que utiliza tensão 3,3v.

5.1.2 Kit Relé duplo para Arduino

É um módulo de 2 Canais de interface de relés, capaz de controlar vários aparelhos e outros equipamentos com corrente de grande porte. Ele pode ser controlado diretamente pelo microcontrolador Arduino.

5.1.3 Protoboard Shield Arduino

Uma protoboard permite que os componentes eletrônicos possam ser interligados de diversas maneiras, para produzir seus circuitos eletrônicos não são necessários nenhum tipo de solda, sendo possível modificar ou rever os circuitos com muita facilidade. As conexões são em bronze banhados a níquel com grampos de mola, podendo ser fixado facilmente através de uma folha adesiva localizada na parte traseira do protoboard.

5.1.4 Fonte de Energia

Fonte de alimentação 12v para o Arduino.

5.1.5 Clock RTC DS3231

É um relógio de tempo real com calendário completo, capaz de fornecer informações de segundo, minutos, dia, data, mês e ano. Podendo operar tanto no formato 12 horas como 24 horas. Em caso de falha de energia o DS3231 automaticamente aciona a bateria que acompanha o módulo para evitar perda de dados.

6 METODOLOGIA

Neste trabalho foi realizado o levantamento das características e operação do sistema embarcado com Arduino, visando seu uso em sistema de pequeno porte. Foi realizado um estudo exploratório de caráter bibliográfico, em que foi possível compreender as situações para o desenvolvimento deste projeto. Para a realização deste trabalho, foi dividido em cinco fases:

a) Fase 1: Levantamento das necessidades escolar, sobre o desenvolvimento de um Sistema Embarcado que acionará o sinal de intervalo entre as aulas.

b) Fase 2: Análise das características de sistemas de pequeno porte, com base nas literaturas e nos resultados que se obtém no desenvolvimento de sistemas embarcados já existentes.

c) Fase 3: Estudo sobre as arquiteturas de *hardware* e *software* para utilização em Sistemas Embarcados.

d) Fase 4: Montagem do *hardware* com Arduino e desenvolvimento da aplicação.

e) Fase 5: Realização de testes e implantação do protótipo.

6.1 Documentação

Uma pesquisa advém de uma metodologia composta de várias fases, desde a formação de um problema até o surgimento dos resultados (GIL, 2009). Apenas se dá início a uma pesquisa se existir um questionamento para a qual se deseja obter uma resposta.

Consequentemente pesquisar é buscar ou procurar respostas para os problemas propostos. Este projeto foi realizado através de pesquisas, para buscar conhecimentos que pudessem resolver o problema escolar.

6.2 Natureza da Pesquisa

Este trabalho mostra o conhecimento por várias pesquisas de teor teórico e ativamente exploratório. Tendo como objetivo a identificação e definição do problema. Contribuindo para todo este primeiro momento de busca e aprendizado científico a compreensão do problema apresentado. Pois toda pesquisa inicial amplia a propagação, gerando novos conceitos (RUIZ, 2009).

6.3 Tipo da Pesquisa

Uma pesquisa engloba toda a literatura que seja proposta ao que se deseja estudar. Compõe-se de publicações, livros, teses, revistas, monografias, pesquisas, etc. Toda pesquisa campo tem o objetivo de abstrair conhecimentos e informações sobre um tipo de problema, empregando atenção aos fatos e ao colhimento de informações concernente ao tema (MARCONI; LAKATOS, 2003).

A pesquisa adotada na elaboração deste trabalho foi baseada em diversas pesquisas bibliográficas de livros, revistas e artigos publicados na internet, buscando comprovar os propósitos para a realização do projeto.

6.4 Técnicas da Pesquisa

As técnicas utilizadas nesta pesquisa foram de cunho tecnológico. Mas para uma ampla compreensão sobre a tecnologia, vejamos a sua definição, pois a palavra tecnologia deriva do grego tecno (*tekhne*) que significa técnica que envolve habilidades entre o saber e o fazer; já *logia* que também é do grego significa estudo (HERSCHBACH, 1995, P.32). Este termo tecnologia está associado ao conhecimento e para se obter conhecimento deve se utilizar da leitura.

A leitura ajuda o pesquisador a alcançar seus objetivos, ler um determinado conteúdo é um exercício muito além da experiência de decifrar signos, mas que se adquire e se torna um hábito ao longo da vida e na qual ele constrói em convivência com a sociedade. Através da leitura amplia-se e aprofunda o conhecimento (FREIRE, 2009)

6.5 Limitações da Pesquisa

No presente projeto existem algumas dificuldades encontradas conforme descrito abaixo:

a) Alguns sites não serem fontes de pesquisas confiáveis;

b) A escola não possuir nobreak a fim de evitar que quando ocorra queda de energia o computador permaneça ainda ligado por mais um tempo;

c) Quando ocorrer algum feriado o funcionário deverá desativar o alarme na tela do sistema para que não ocorram acionamentos;

d) O sistema operacional padrão das escolas públicas serem Linux, e a aplicação desenvolvida é com funcionamento em ambiente Windows. Nas escolas possuem alguns computadores na plataforma Windows.

7 TRABALHOS CORRELATOS

O projeto Campainha escolar com sinalização de emergência, visa à construção de um acionamento escolar, mas também com um plano de emergência. Neste projeto foi utilizado como base o microcontrolador PICAXE-28X1 num circuito integrado DS1307 com relógio em tempo real, onde o usuário irá realizar as configurações através de teclas e sua visualização se dará através de um *display*.

O projeto Sistema Microcontrolado de Acionamento de Sirene Escolar – SMASE foi desenvolvido para o controle de uma sirene escolar, onde é composto por um microcontrolador PIC18F2550, que permite ao usuário o controle da sirene em tempo real. O *software* supervisório do sistema é desenvolvido na linguagem Delphi.

Os projetos acima citados possuem um papel central e correlacionado em ambos, embora o enfoque seja distinto. Considerando que se trata de trabalhos sobre temas correlatos e as pesquisas envolvidas possuam desenvolvimentos semelhantes, pode se esperar melhorias e atributos na construção de cada novo projeto a ser formado.

8 CRONOGRAMA

Conforme quadro abaixo segue cronograma das atividades realizadas para a elaboração deste projeto.

Quadro 3: Cronograma das atividades

Revisão Literária	Fev	Mar	Abr	Mai	Jul	Ago	Set	Out	Nov	Dez
Pesquisas										
Pesquisa Bibliográficas	X	X	X							
Definição dos objetivos e funcionalidades	X	X								
Definição das Ferramentas do Projeto	X	X	X							
Elaboração do Projeto			X							
Entrega do TCC I				X						
Defesa de Banca TCC I					X					
Adequação da Proposta para criação de projetos (TCCII)						X				
Definição das funcionalidades do sistema							X	X		
Execução do Projeto (Software)									X	
Testes									X	
Implantação										X
Entrega do TCC II									X	
Defesa de Banca TCC II										X

Fonte: Arquivo do Autor

9 PROJETO

Neste item, a definição e a estrutura do projeto, assim como, ele foi realizado e sobre a interface do usuário, simultaneamente com as funcionalidades do protótipo, apresentando os dados para seu funcionamento. Foi realizada a estrutura de desenvolvimento do sistema embarcado na plataforma Arduino e as funcionalidades definidas do sistema.

9.1 Hardware

Para a implantação do projeto que foi desenvolvido, serão realizados dois passos, que será a conexão do *hardware* e a instalação do sistema.

O primeiro passo será a conexão via USB da placa Arduino no computador a ser definido pela escola, e a ligação da sirene no *hardware* conectado ao computador.

O segundo passo é a instalação do sistema no computador, que deve possuir sistema operacional Windows. Não haverá restrição quanto à quantidade de espaço, pois toda a configuração de horários do sistema irá estar dentro de um Micro SD, no entanto não haverá necessidade de estar conectado todo o tempo no computador, pois o mesmo é um sistema embarcado. Após a configurações dos horários o mesmo poderá ser desconectado do computador, realizando os acionamentos normalmente. A comunicação ocorrerá através de um cabo *Ethernet*, que ligará o computador com a plataforma Arduino. Quando o Arduino estiver desligado do computador o mesmo deve ser ligado a uma fonte de 9v ou 12v.

9.2 Telas do Sistema Embarcado

Conforme a necessidade apresentada pelo ambiente escolar será apresentada as interfaces do sistema, para explanar de forma visual como irá ocorrer o seu funcionamento, e como será a utilização pelo usuário. O sistema possui uma interface bem fácil para realizar as funções propostas, tornando assim a experiência do usuário com o sistema de uma forma bem compreensível.

Figura 7: Tela Inicial com Usuário e Senha

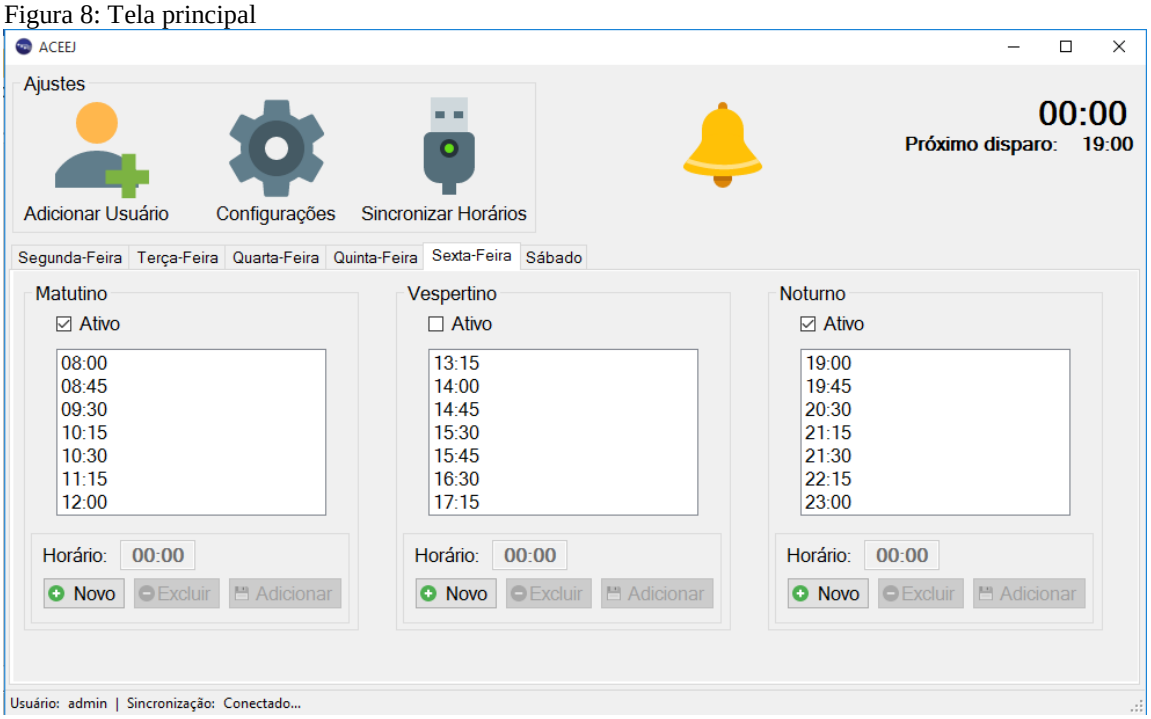
Fonte: Arquivo do Autor

Na figura 7, compõe se da tela inicial, com solicitação de usuário e senha, para acesso ao sistema. Por padrão o usuário e senha inicial serão (admin/admin).

Na figura 8 é apresentada a interface da segunda tela. Após ter sido informado o usuário e a senha será possível incluir um novo usuário no botão Adicionar Usuário, no botão Configurar será possível selecionar a porta USB e alterar o tempo de disparo da sirene, no botão Sincronizar Horários será possível enviar os horários cadastrados para o Arduino e seguidamente o mesmo será armazenado no cartão SD.

Nesta mesma tela possui várias abas referente aos dias da semana que listará os horários cadastrados pelo usuário e um checkbox para ativar ou desativar os acionamentos em cada dia da semana.

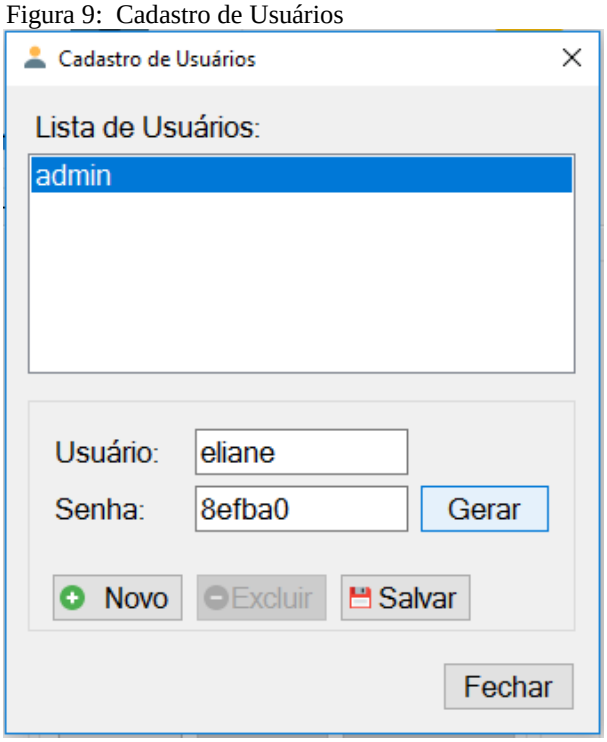
Figura 8: Tela principal



Fonte: Arquivo do Autor

Nesta tela podem-se configurar os horários desejados, excluindo ou alterando os horários do intervalo das aulas, horário de início e termino das aulas e selecionar os dias da semana que deseja que o sinal seja ativado, após a realização das configurações basta apenas salvar.

Figura 9: Cadastro de Usuários

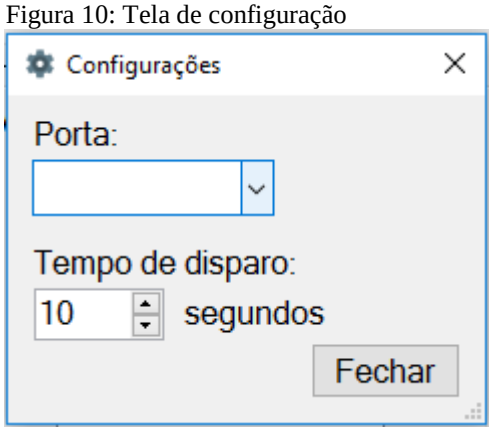


Fonte: Arquivo do Autor

A figura 9 representa a tela de cadastro de usuários que irá aparecer após clicar no botão adicionar usuário. Poderá ser adicionado um novo usuário e ser gerada senha para cada usuário.

A figura 10 representa a tela de configuração onde será selecionada a porta USB conectada e o tempo de disparo da sirene.

Figura 10: Tela de configuração



Fonte: Arquivo do Autor

9.2.2 Interface

Conforme foi mencionado no referencial teórico, o sistema foi desenvolvido na linguagem C#, com uma aplicação do tipo *Windows Forms*. O mesmo deve ser instalado no computador local da escola, para realizar a interação do usuário com o Arduino. Essa comunicação ocorrerá através de um cabo *Ethernet*, que ligará o computador com a plataforma Arduino.

Para ter acesso a esse sistema, o usuário deverá efetuar o *login* com uma senha, o gerenciamento de usuários e senhas ficará no próprio Arduino, em uma memória Micro SD, este terá um arquivo com os usuários e suas permissões, para que apenas estes usuários possam controlar e agendar os horários de utilização do sistema embarcado para controle do sinal de intervalo.

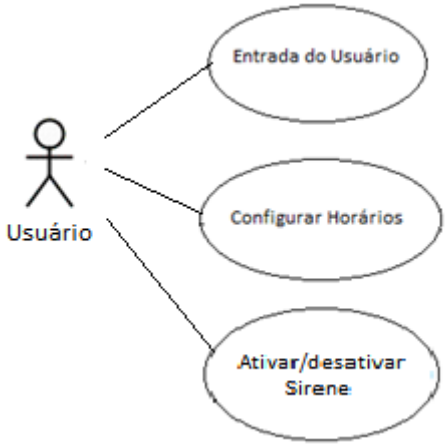
O sistema irá acionar a sirene na troca de horários, considerando que não efetuará o acionamento nos finais de semana, possíveis feriados e férias escolares, de acordo com as configurações efetuadas pelo usuário. A interface do sistema é de simples utilização.

9.3 Modelagem Ágil

A utilização de modelagem facilita a compreensão de um problema e respectivamente de sua solução. A Modelagem Ágil é fundamentada numa prática de modelagem eficiente (AMBLER, 2014).

A Modelagem ágil tem como propósito de conduzir à prática em sua totalidade a junção de valores e princípios pertinentes a uma modelagem leve e poderosa (AMBLER, 2004).

Figura 11: Diagrama de Caso de Uso

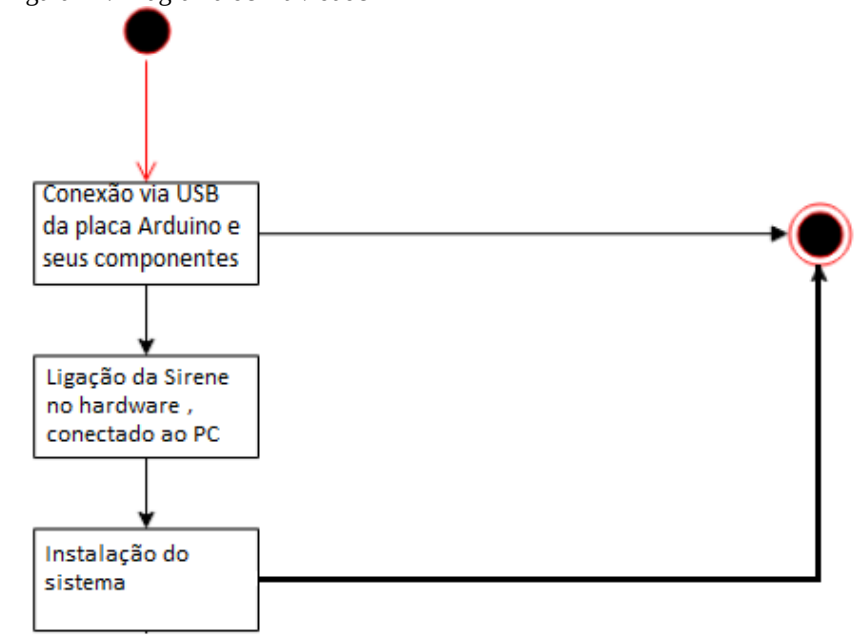


Fonte: Arquivo do Autor

9.3.1 Diagrama de Atividade

Um diagrama de atividade demonstra o processo de um *software* ou um fluxo de trabalho, descrevendo uma série de ações. Serão demonstradas as ações que ocorrerão em certas situações simulando o processo que ocorre dentro do sistema (MICROSOFT, 2016).

Figura 12: Diagrama de Atividade

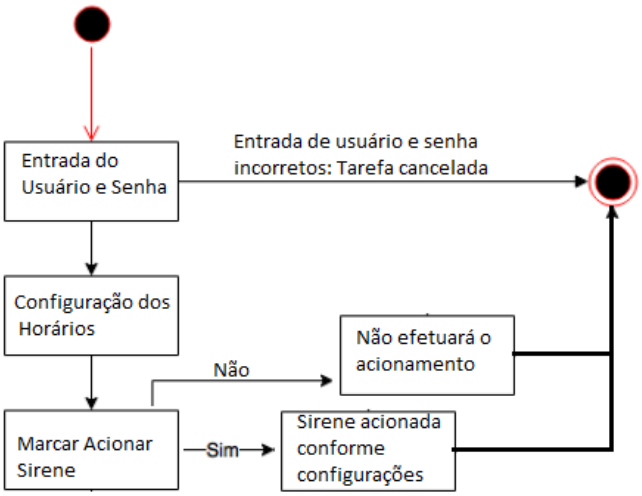


Fonte: Arquivo do Autor

9.3.2 Diagrama de atividade do Fluxo de interface com o Usuário

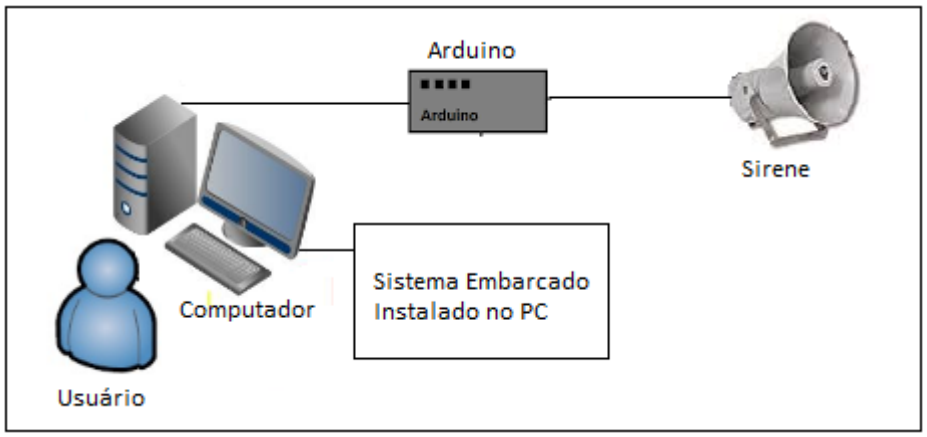
Após os estudos de referencial teórico foi elaborado um diagrama de fluxo de interface com o usuário que dará uma visão mais ampla de como será desenvolvido o projeto.

Figura 13: Diagrama de atividade do fluxo de interface com o usuário



Fonte: Arquivo do Autor

Figura 14: Diagrama de Fluxo da montagem

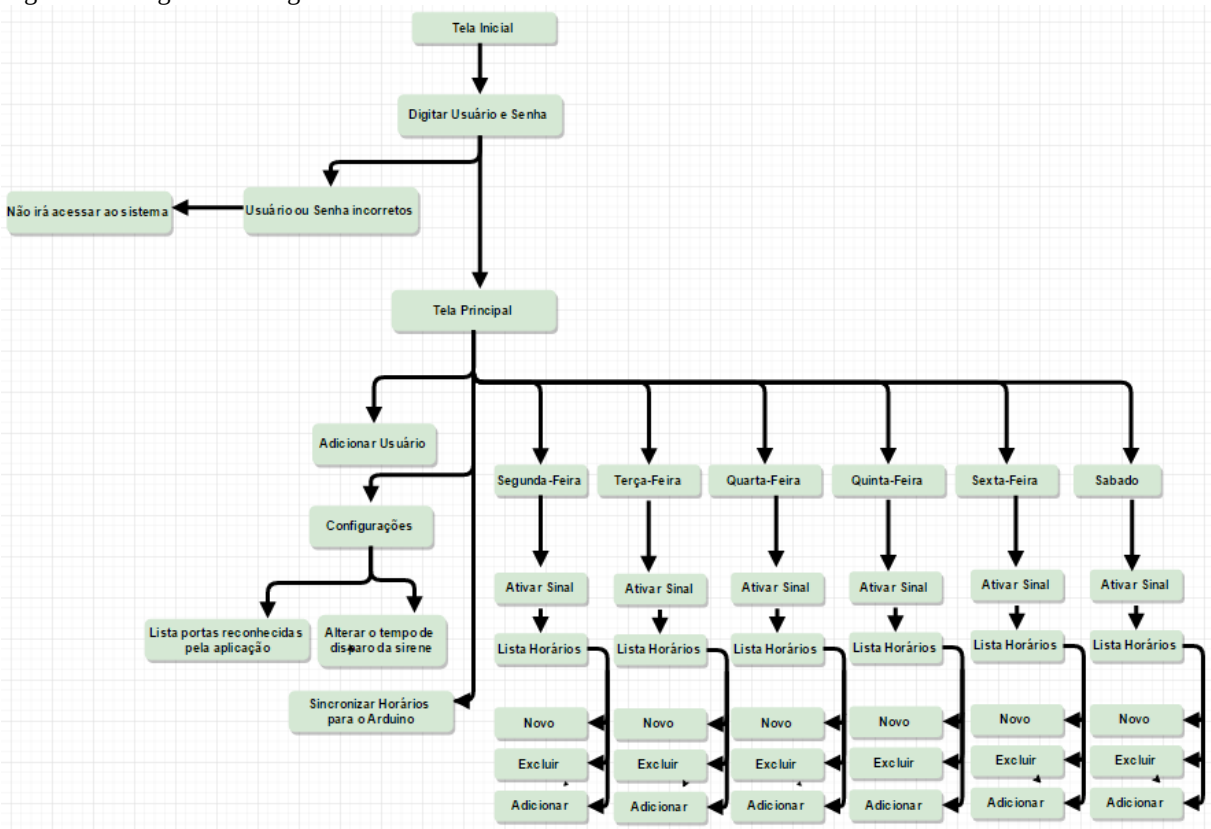


Fonte: Arquivo do Autor

9.3.3 Diagrama Navegacional OOHDM

Foi elaborado o Diagrama Navegacional OOHDM que é orientado a objetos onde demonstra toda a estrutura navegacional da aplicação.

Figura 15: Diagrama Navegacional OOHDM



Fonte: Arquivo do Autor

10 RESULTADOS

As pesquisas realizadas neste trabalho demonstram que os sistemas embarcados estão cada vez mais presentes em vários equipamentos eletrônicos, são dispositivos que estão envoltos de equipamentos maiores e acabam por ser impercebíveis aos usuários.

O desenvolvimento deste projeto tem como finalidade disponibilizar à escola a primeira versão do ACEEJ que será para uso gratuito no ambiente escolar. Está aplicação desenvolvida em sistema embarcado utilizando a plataforma Arduino pode auxiliar no gerenciamento e controle dos sinais de intervalos escolares, pois é uma evolução dos antigos padrões de acionamento de sirene escolar, é ela que irá informar o início, o intervalo e o final das aulas.

Antes era necessário que um funcionário fosse até o local específico da sirene para que a mesma fosse acionada. Com o ACEEJ é possível que ocorra os acionamentos de forma automática nos horários programados, evitando consideravelmente falhas e atrasos no acionamento da sirene devido a utilizar um sistema embarcado, basta apenas programar os horários dos acionamentos e sua duração.

REFERÊNCIAS

AMBLER, S. W. **Modelagem Ágil:** Práticas eficazes para a Programação Extrema e o Processo Unificado. Porto Alegre: Bookman, 2004. Disponível em: <https://goo.gl/aSuUc1> Acesso em: 12 março 2016.

AMBLER, S. W. **Agile Modeling.** 2014. Disponível em <http://www.agilemodeling.com/> Acesso em: 06 maio 2016.

ARDUINO.**Whats is Arduino?**
Disponível em: <https://www.arduino.cc/en/Guide/Introduction#> Acesso em: 26 março 2016.

ARDUINO. **Arduino Uno.**
Disponível em <https://www.arduino.cc/en/Main/arduinoBoardUno> Acesso em: 06 maio 2016.

AUTODESK. **Autodesk 123 Circuits Online.** Disponível em <https://123d.circuits.io/lab> Acesso em: 28 maio 2016.

AKYILDIZ et al. 2002. **Wireless sensor networks:** a survey. Computer networks, 38(4):393–422. Disponível em: <http://www.ee.oulu.fi/~carlos/WSNPapers/AK02.pdf> Acesso em: 05 março 2016.

BANNATYNE, R. **Don't write-off 8-bit microcontrollers**. Agosto de 2009. Disponível em: <http://www.electronicsweekly.com/news/products/micros/dont-write-off-8-bit-microcontrollers-2009-08/> Acesso em: 20 março 2016.

BARRAGÁN, H. (2012). **Wiring**. Disponível em: <http://wiring.org.co> Acesso em: 27 março 2016.

BRAGA, C. N. (2001). **Microcontrolador MSP430 – Parte III (MIC094).** Disponível em: <http://www.newtoncbraga.com.br/index.php/microcontroladores/142-texas-instruments/8217-microcontrolador-msp430-parte-iii-mic094> Acesso em: 26 Março 2016.

BERGER, A. **Embedded Systems Design:** An Introduction to Processes, Tools, and Techniques. CMP Books, 2002. Ed. Taylor & Francis, 2002. Disponível em: < https://books.google.com.br/books/about/Embedded_Systems_Design.html?id=3vY35UkvXrAC&redir_esc=y> Acesso em: 23 abril 2016.

BUTTAZZO, G. **Research trends in real-time Operating Systems**. ACM SIGBED Rewiew, Julho de 2006. Disponível em: < http://retis.sssup.it/~giorgio/paps/2006/ACM-review06.pdf> Acesso em: 06 abril 2016.

CARRO, L.; WAGNER, F. R. **Metodologias e Técnicas de engenharia de software para Sistemas Embarcados,** Jornadas de Atualização em Informática da SBC, 2003.

CARVALHO, C. **Campainha escolar com sinalização de emergência.** Escola Secundária Afonso Lopes Vieira Curso Profissional de Técnico de Eletrônica e Telecomunicações 2009/2012. Disponível em: < http://esalvieira-m.ccems.pt/file.php/1/paulo_santos/pap_cptet2012/CPTET2012_RelatPAP_04CarlaCarvalho_Final.pdf> Acesso em 28 maio 2016.

CHASE, O. A. **Sistemas Embarcados**. Artigo Técnico de Introdução a Tecnologia dos Sistemas Embarcados, 2007. Disponível em: < http://www.lyfreitas.com.br/ant/pdf/Embarcados.pdf> Acesso em: 28 março 2016

CULKIN, J. (2012). **Arduino**. Disponível em: <http: //www.jodyculkin.com/wp-content/uploads/2012/04/arduino-comic-latest.pdf> Acesso em: 27 março 2016.

CUNHA, A. F., **O que são Sistemas Embarcados**. Revista Saber Eletrônica, Edição 414, Editora Saber, São Paulo, 2007.

DELAI, L. A. **Sistemas Embarcados: a computação invisível**. Disponível em: <http://www.hardware.com.br/artigos/sistemas-embarcados-computacao-invisivel/conceito.html> Acesso em: 28 de março de 2016.

EMBARCADOS (2013). **Sistema Embarcado** – **O que é? Qual a sua importância?** Disponível em: <http://www.embarcados.com.br/sistema-embarcado/> Acesso em: 28 março 2016.

EMBARCADOS. **Linguagens para Sistemas Embarcados.** Disponível em: <http://www.embarcados.com.br/editorial-linguagens-para-sistemas-embarcados/> Acesso em 11 abril 2016.

FILHO SILVA, A. **Engenharia de software** – Requisitos não funcionais – Critérios para análise arquitetural. Disponível em: <http://www.devmedia.com.br/artigo-engenharia-de-software-3-requisitos-nao-funcionais/9525> Acesso em 08 abril 2016.

FREIRE, P. **A Importância do ato de ler:** em três artigos que se completam. 50. ed. São Paulo: Cortez, 2009. P. 87, p. 1-87. ISBN 9788524903083.

GANSLLE. J.G. **The Art of Designing Embedded Systems**. 2.ed. U.S.A.: Elsevier, 2008. 298 p., il. ISBN 9780750686440.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2009. 175 p. ISBN 9788522431694.

HAMACHER et al. 2012. **Computer organization and embedded systems**. McGraw-Hill. 6 ed. P. 735. P.385-417. ISBN 978-0-07-338065-0

HENZINGER, T. A., SIFAKIS, E. J. **The Embedded Systems Design Challenge**. Formal Methods (Springer), 2006. Disponível em: < http://www-verimag.imag.fr/~sifakis/RecentPublications/main.pdf> Acesso em: 12 abril 2016

HERSCHBACH, D. R. **Technology as Knowledge:** Implications for Instruction. Journal of Technology Education**.** Vol. 7 No. 1. 1995. Disponível em: <http://scholar.lib.vt.edu/ejournals/JTE/v7n1/pdf/herschbach.pdf>. Acesso em: 04 maio 2016.

LEE, E. A. **Embedded Software**. Advances in Computers Vol. 56, 2002: Pg. 55-95.

LI, Q. **Real Time concepts for Embedded Systems**. CMP Books, 2003. Disponível em: < https://books.google.com.br/books?id=6xh-CwAAQBAJ> Acesso em: 23 março 2016

MAGNUN. Hack 'n' Cast. Hack ‘n’ Cast v0.15 - **Sistemas Embarcados - Parte 1**. Disponível em: <http://mindbending.org/pt/hack-n-cast-v015-sistemas-embarcados-parte-1> Acesso em: 29 março 2016.

MARCONI, M. A.; LAKATOS, E. M. **Fundamentos de metodologia cientifica.** 5. ed. São Paulo: Atlas, 2003.

MCROBERTS, M. **Arduino Básico**. 3 reimp. P. 5-68 Editora: Novatec, São Paulo, 2011.

MICROSOFT. **Diagramas de atividade UML:** referência. Disponível em <https://msdn.microsoft.com/pt-br/library/dd409360.aspx> Acesso em: 07 abril 2016.

MORIMOTO, C. E. **Entendendo os sistemas embarcados**. Fevereiro de 2007. Disponível em: <http://www.hardware.com.br/artigos/entendendo-sistemas-embarcados/> Acesso em: 29 março 2016.

MULTILÓGICA-SHOP. **Arduino Uno R3**. Disponível em: <https://multilogica-shop.com/arduino-uno-r3> Acesso em 05 maio 2016.

MURPHY, C. **This New House.** Setembro de 2005. Disponível em: <http://archive.fortune.com/magazines/fortune/fortune_archive/2005/09/19/8272889/index.htm> Acesso em: 20 março 2016.

NOERGAARD, T. **Embedded System Architecture** – A comprehensive Guide for Engineers and Programmers. Elsevier, 2005.Disponível em: < http://goo.gl/atg7cF > Acesso em: 20 março 2016.

ORTIZ JR. S. **Embedded OSs Gain the Inside Track**. Computer (IEEE Computer Society) Vol.34, n. Issue 11 Novembro 2001. Disponível em: < http://dl.acm.org/citation.cfm?id=621831&CFID=796018506&CFTOKEN=65798193 > Acesso em: 01 março 2016

PINTO, PEDRO. (2013). **S4A – Programar o Arduino é fácil e divertido.** Disponível em: <http://pplware.sapo.pt/gadgets/hardware/s4a-programar-o-arduino-e-facil-e-divertido/> Acesso em: 12 abril 16.

RUIZ, J. Á. **Metodologia científica:** guia para eficiência nos estudos. 6 ed. São Paulo: Atlas, 2009 180 p. ISBN 9788522444823.

SCRATCH. **About Schratch.** Disponível em: <https://scratch.mit.edu/about/> Acesso em: 12 abril 2016.

SILVA, A. **Conheça o Arduino Mega 2560, a mais poderosa placa Arduino**. Disponível em: <http://www.oarduino.com/conheca-o-arduino-mega-2560/> Acesso em: 06 abril 2016.

SOUSA, D. (2012). **Ardublock: programação em blocos para Arduino.** Disponível em: <http://www.portalmcu.com.br/2013/12/ardublock-programacao-em-blocos-para-arduino.html> Acesso em: 12 abril 2016

SOUZA, F. (2014). **Arduino – Ardublock – Programação visual para Arduino.** Disponível em: <http://www.embarcados.com.br/arduino-ardublock/> Acesso em: 12 abril 2016.

SOUZA, F. (2014). **Comunicação Serial com C# e Arduino – Parte 2.** Disponível em: <http://www.embarcados.com.br/comunicacao-serial-c-arduino-parte-2/> Acesso em: 12 abril 2016.

SOUZA, F. **Arduino Mega 2560.** Disponível em: <http://www.embarcados.com.br/arduino-mega-2560/> Acesso em: 02 abril 2016.

SOUZA, et al. **Sistema microcontrolado de acionamento de sirene escolar - SMASE.** 2011. VI Congresso de Pesquisa e Inovação da Rede Norte e Nordeste de Educação Tecnológica Aracaju-SE -2011. Disponível em: <http://www.ebah.com.br/content/ABAAAfAOsAK/sistema-microcontrolado-acionamento-sirene-escolar-smase>. Acesso em: 26 maio 2016.

STALLINGS, W. **Operating System:** Internals and Design Principles. 7. Ed. P. 573-606 Boston: Pearson Prentice Hall, 2012 768 p. ISBN 9780132309981.

SUEIRO, D. **Uma pessoa interage com 20 sistemas embarcados por dia.** Agosto de 2014. Disponível em: <http://www.escbrazil.com.br/pt/multimedia/indice-de-noticias/382-uma-pessoa-interage-com-20-sistemas-embarcados-por-dia> Acesso em: 20 março 2016.

S4A. **About S4A.** Disponível em: <http://s4a.cat/>. Acesso em: 12 abril 2016.

TANENBAUM, A. S. **Sistemas operacionais modernos**. Tradução de Ronaldo A. L. Gonçalves, Luís A. Consularo, Luciana do Amaral Teixeira. 3. ed. P. 43-45;615-617, São Paulo: Pearson Prentice Hall, 2009. 653 p., il. ISBN 9788576052371.

WIKIMEDIA. **File Minuteman.**
Disponível em: <https://commons.wikimedia.org/wiki/File:Minuteman_I.jpg> Acesso em 27 março 2016.

WOLF. W. **Computers as Components** – Principles of Embedded Computing System Design. Morgan Kaufmann Publishers, San Francisco, 2. ed. USA: Elsevier, c2008. 507 p., il. ISBN 9780123743978.

APÊNDICE A – Código Fonte do Arduino

```
#include <DS3231.h>
//#include <DS1307.h>
#include <SD.h>
// Init the DS3231 using the hardware interface
DS3231 rtc(SDA, SCL);
//DS1307 rtc(SDA, SCL);

File myFile;

const unsigned int MAX_INPUT = 400;
const int chipSelect = 4;
String lastDayWeek = "";
String line = "";
int shotTime = 5;
String lastTime = "";

void(* resetFunc) (void) = 0;

void setup() {
  Serial.begin(9600);
  while (!Serial) {}

  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
  digitalWrite(5, HIGH);

  Serial.print("Initializing SD card...");

  if (!SD.begin(chipSelect))
  {
    //sd.initErrorHalt();
    //, SPI_HALF_SPEED
  }
  Serial.println("card initialized.");

  Serial.print("Initializing RTC...");
  rtc.begin();
   // Set the clock to run-mode
  //rtc.halt(true);

  lastDayWeek = rtc.getDOWStr();
  readFile(lastDayWeek);
  Serial.println("RTC initialized.");
}

void setConfiguration(byte incomming) {
```

```
  if (SD.exists("CONFIG.TXT")) {
    SD.remove("CONFIG.TXT");
  }

  char inputline[MAX_INPUT];
  unsigned int inputpos = 0;

  while (Serial.available())
  {
    char inByte = Serial.read();
    switch (inByte)
    {
      case '\n':
        {
          inputline [inputpos] = 0;

          myFile = SD.open("CONFIG.TXT", FILE_WRITE);

          myFile.println(inputline);
          myFile.close();

          inputpos = 0;
        }
        break;
      case '\r':
        break;

      default:
        {
          if (inputpos < (MAX_INPUT - 1))
            inputline [inputpos++] = inByte;
        }
        break;
    }//end-switch
    delay(50);
  }
  Serial.println("d|true");
  resetFunc();  //call reset
}

void readFile(String dayOfWeekStr) {
  myFile = SD.open("CONFIG.TXT");

  char input_line[MAX_INPUT];
  unsigned int input_pos = 0;
  byte data;
  boolean isDay = false;
  boolean proc = true;
  String inputLineStr = "";
```

```
  Serial.println("while");
  shotTime = myFile.readStringUntil('\n').toInt();
  while (((int)(data = myFile.read()) != 255) && proc == true) {
    switch (data)
    {
      case '\n':
        {
          if (isDay) {
            proc = false;
          }
          else {
            input_pos = 0;
            memset(input_line, 0, sizeof(input_line));
          }
        }
        break;
      case '\r': break;
      case '|':
        {
          inputLineStr = String(input_line);
          String str = inputLineStr.substring(0, dayOfWeekStr.length());

          if (dayOfWeekStr == str) {
            Serial.println(str);
            isDay = true;
          }
          inputLineStr = "";
          input_pos = 0;
          memset(input_line, 0, sizeof(input_line));
        }
        break;
      default:
        if (input_pos < (MAX_INPUT - 1))
          input_line[input_pos++] = data;
        break;
    }
  }
  myFile.close();
  Serial.println(input_line);
  line = input_line;
}

void comparaLinha(char * currentTime) {
  if ((line.indexOf(currentTime) > -1) && (lastTime.indexOf(currentTime) < 0)) {
    digitalWrite(5, LOW);
    delay(shotTime * 1000);
    digitalWrite(5, HIGH);
    lastTime = currentTime;
    Serial.println("Disparar");
  }
```

```
}

void setRTCDateTime() {
  String dateTime = "";
  if (Serial.available() > 0)
  {
    dateTime = Serial.readStringUntil('\n');
    int hh = dateTime.substring(0, 2).toInt();
    int mm = dateTime.substring(3, 5).toInt();
    int ss = dateTime.substring(6, 8).toInt();

    int dd = dateTime.substring(9, 11).toInt();
    int mth = dateTime.substring(12, 14).toInt();
    int yy = dateTime.substring(15, 19).toInt();

    int dayWeek = dateTime.substring(20, 21).toInt();

    rtc.setTime(hh, mm, ss);
    rtc.setDate(dd, mth, yy);
    rtc.setDOW(dayWeek);
  }
}

void loop() {
  //delay (1000);
  if (Serial.available()) {
    byte incomming = Serial.read();

    switch (incomming)
    {
      case 'd':
        {
          setConfiguration(incomming);
          lastDayWeek = "";
          lastTime = "";
          break;
        }
      case 't':
        {
          setRTCDateTime();
          lastTime = "";
          Serial.println("t|true");
          break;
        }
    }
  }

  //Serial.println("Comparando dia da semana");
  if (lastDayWeek != rtc.getDOWStr()) {
    resetFunc();
```

```
  }

  char *currentTime = rtc.getTimeStr(FORMAT_SHORT);
  Serial.println(line);
  Serial.println(currentTime);

  comparaLinha(currentTime);

  digitalWrite(6, HIGH);
  delay (1000);
  digitalWrite(6, LOW);
}
```

APÊNDICE B – Classe ArduinoController de sincronização entre o computador e o Arduino

```
public class ArduinoController
   {
      private SerialPort _serial;
      private string _port;
      private DateTime _clock;

      public DateTime Clock
      {
         get { return _clock; }
         set { _clock = value; }
      }

      public ArduinoController(string port)
      {
         try
         {
            _port = port;
            _serial = new SerialPort(port, 9600);
         }
         catch (Exception)
         {
            throw new Exception("Dispositivo desconectado ou configurado na porta errada");
         }
      }

      public void Open()
      {
         try
         {
            if (_serial.IsOpen)
```

```
            return;

        _serial.Open();
        _serial.DataReceived += serial_DataReceived;
    }
    catch (Exception)
    {
        throw new Exception("Dispositivo desconectado ou configurado na porta errada");
    }
}

public void Close()
{
    _serial.Close();
}

private void serial_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    try
    {
        var data = _serial.ReadLine();
        Console.WriteLine(data);
        DateTime hrDisp;

        if (DateTime.TryParse(data, out hrDisp))
            Principal.Instance.HourArduino(hrDisp);
        else
        {
            var result = data.Split('|');

            switch (result[0])
            {
                case "d":
                    SetConfig(data);
                    break;

                case "t":
                    Console.WriteLine("Configurado");
                    break;
            }
        }
    }
    catch (Exception)
    {
        Principal.Instance.SetMessageFoot("Mensagem de resposta desconhecida");
    }
}

private void SetConfig(string data)
{
```

```
            if (data.Contains("true"))
                Principal.Instance.SetMessageFoot("Configuração enviada com sucesso");
        }

        public bool IsOpen { get { return _serial.IsOpen; } }

        public void SendFileConfig(ConfigurationEntity config)
        {
            var cfg = config.Schedules.Where(s => s.Active);
            _serial.Write("d");
            _serial.WriteLine(config.ShotTime.ToString());
            foreach (var item in cfg)
            {
                var charoles = item.ToString().ToCharArray();
                for (int i = 0; i < charoles.Length; i++)
                {
                    _serial.Write(charoles[i].ToString());
                    Thread.Sleep(45);
                }
                _serial.WriteLine(string.Empty);
                Thread.Sleep(110);
            }
        }

        public void SetClock()
        {
            try
            {
                if (!_serial.IsOpen)
                    throw new Exception("Dispositivo desconectado ou configurado na porta
errada");

                _serial.WriteLine(string.Format("t{0}T{1}|{2}"
                , DateTime.Now.ToLongTimeString()
                , DateTime.Now.ToShortDateString()
                , (int)DateTime.Now.DayOfWeek));
            }
            catch (Exception)
            {
                throw;
            }
        }
    }
}
```

APÊNDICE C – Classe HandleControl.Designer – Classe onde foi desenvolvida a maioria das funcionalidades da aplicação

```
partial class HandleControl
   {
      /// <summary>
      /// Required designer variable.
      /// </summary>
      private System.ComponentModel.IContainer components = null;

      /// <summary>
      /// Clean up any resources being used.
      /// </summary>
      /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
      protected override void Dispose(bool disposing)
      {
         if (disposing && (components != null))
         {
            components.Dispose();
         }
         base.Dispose(disposing);
      }

      #region Component Designer generated code

      /// <summary>
      /// Required method for Designer support - do not modify
      /// the contents of this method with the code editor.
      /// </summary>
      private void InitializeComponent()
      {
         this.components = new System.ComponentModel.Container();
         this.tableLayoutPanel1 = new System.Windows.Forms.TableLayoutPanel();
         this.panel1 = new System.Windows.Forms.Panel();
         this.lblNextShot = new System.Windows.Forms.Label();
         this.lblTextNextShot = new System.Windows.Forms.Label();
         this.groupBox1 = new System.Windows.Forms.GroupBox();
         this.picSend = new System.Windows.Forms.PictureBox();
         this.label1 = new System.Windows.Forms.Label();
         this.picConfig = new System.Windows.Forms.PictureBox();
         this.label2 = new System.Windows.Forms.Label();
         this.picAddUser = new System.Windows.Forms.PictureBox();
         this.lblAddUser = new System.Windows.Forms.Label();
         this.lblTimer = new System.Windows.Forms.Label();
         this.pictureBox3 = new System.Windows.Forms.PictureBox();
         this.tbcSchedules = new System.Windows.Forms.TabControl();
         this.tpbMonday = new System.Windows.Forms.TabPage();
```

```
            this.tpgTuesday = new System.Windows.Forms.TabPage();
            this.tpgWednesday = new System.Windows.Forms.TabPage();
            this.tpgThursday = new System.Windows.Forms.TabPage();
            this.tpgFriday = new System.Windows.Forms.TabPage();
            this.tpgSaturday = new System.Windows.Forms.TabPage();
            this.timerClock = new System.Windows.Forms.Timer(this.components);
            this.tableLayoutPanel1.SuspendLayout();
            this.panel1.SuspendLayout();
            this.groupBox1.SuspendLayout();
            ((System.ComponentModel.ISupportInitialize)(this.picSend)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.picConfig)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.picAddUser)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox3)).BeginInit();
            this.tbcSchedules.SuspendLayout();
            this.SuspendLayout();
            //
            // tableLayoutPanel1
            //
            this.tableLayoutPanel1.ColumnCount = 1;
            this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 100F));
            this.tableLayoutPanel1.Controls.Add(this.panel1, 0, 0);
            this.tableLayoutPanel1.Controls.Add(this.tbcSchedules, 0, 1);
            this.tableLayoutPanel1.Dock = System.Windows.Forms.DockStyle.Fill;
            this.tableLayoutPanel1.Location = new System.Drawing.Point(0, 0);
            this.tableLayoutPanel1.Name = "tableLayoutPanel1";
            this.tableLayoutPanel1.RowCount = 2;
            this.tableLayoutPanel1.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 150F));
            this.tableLayoutPanel1.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 100F));
            this.tableLayoutPanel1.Size = new System.Drawing.Size(854, 525);
            this.tableLayoutPanel1.TabIndex = 0;
            //
            // panel1
            //
            this.panel1.Controls.Add(this.lblNextShot);
            this.panel1.Controls.Add(this.lblTextNextShot);
            this.panel1.Controls.Add(this.groupBox1);
            this.panel1.Controls.Add(this.lblTimer);
            this.panel1.Controls.Add(this.pictureBox3);
            this.panel1.Dock = System.Windows.Forms.DockStyle.Fill;
            this.panel1.Location = new System.Drawing.Point(3, 3);
            this.panel1.Name = "panel1";
            this.panel1.Size = new System.Drawing.Size(848, 144);
            this.panel1.TabIndex = 0;
            //
            // lblNextShot
            //
```

```
            this.lblNextShot.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Right)));
            this.lblNextShot.AutoSize = true;
            this.lblNextShot.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.lblNextShot.Location = new System.Drawing.Point(791, 55);
            this.lblNextShot.Name = "lblNextShot";
            this.lblNextShot.Size = new System.Drawing.Size(54, 20);
            this.lblNextShot.TabIndex = 13;
            this.lblNextShot.Text = "00:00";
            //
            // lblTextNextShot
            //
            this.lblTextNextShot.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Right)));
            this.lblTextNextShot.AutoSize = true;
            this.lblTextNextShot.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.lblTextNextShot.Location = new System.Drawing.Point(639, 55);
            this.lblTextNextShot.Name = "lblTextNextShot";
            this.lblTextNextShot.Size = new System.Drawing.Size(146, 20);
            this.lblTextNextShot.TabIndex = 12;
            this.lblTextNextShot.Text = "Próximo disparo: ";
            //
            // groupBox1
            //
            this.groupBox1.Controls.Add(this.picSend);
            this.groupBox1.Controls.Add(this.label1);
            this.groupBox1.Controls.Add(this.picConfig);
            this.groupBox1.Controls.Add(this.label2);
            this.groupBox1.Controls.Add(this.picAddUser);
            this.groupBox1.Controls.Add(this.lblAddUser);
            this.groupBox1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.groupBox1.Location = new System.Drawing.Point(3, 3);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size = new System.Drawing.Size(452, 140);
            this.groupBox1.TabIndex = 11;
            this.groupBox1.TabStop = false;
            this.groupBox1.Text = "Ajustes";
            //
            // picSend
            //
            this.picSend.Image = global::ACEEJ.WinForm.Properties.Resources.USB_On_96;
            this.picSend.Location = new System.Drawing.Point(332, 19);
            this.picSend.Name = "picSend";
            this.picSend.Size = new System.Drawing.Size(90, 90);
            this.picSend.SizeMode = System.Windows.Forms.PictureBoxSizeMode.CenterImage;
```

```
            this.picSend.TabIndex = 11;
            this.picSend.TabStop = false;
            this.picSend.Click += new System.EventHandler(this.picSend_Click);
            this.picSend.MouseEnter += new System.EventHandler(this.SetStretchImage);
            this.picSend.MouseLeave += new System.EventHandler(this.SetDefaultImage);
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label1.Location = new System.Drawing.Point(297, 112);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(152, 20);
            this.label1.TabIndex = 12;
            this.label1.Text = "Sincronizar Horários";
            //
            // picConfig
            //
            this.picConfig.Image = global::ACEEJ.WinForm.Properties.Resources.Settings_96;
            this.picConfig.Location = new System.Drawing.Point(183, 19);
            this.picConfig.Name = "picConfig";
            this.picConfig.Size = new System.Drawing.Size(90, 90);
            this.picConfig.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.CenterImage;
            this.picConfig.TabIndex = 9;
            this.picConfig.TabStop = false;
            this.picConfig.Click += new System.EventHandler(this.picConfig_Click);
            this.picConfig.MouseEnter += new System.EventHandler(this.SetStretchImage);
            this.picConfig.MouseLeave += new System.EventHandler(this.SetDefaultImage);
            //
            // label2
            //
            this.label2.AutoSize = true;
            this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label2.Location = new System.Drawing.Point(172, 112);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(112, 20);
            this.label2.TabIndex = 10;
            this.label2.Text = "Configurações";
            //
            // picAddUser
            //
            this.picAddUser.Image =
global::ACEEJ.WinForm.Properties.Resources.Gender_Neutral_User_96;
            this.picAddUser.Location = new System.Drawing.Point(28, 19);
            this.picAddUser.Name = "picAddUser";
            this.picAddUser.Size = new System.Drawing.Size(90, 90);
```

```
            this.picAddUser.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.CenterImage;
            this.picAddUser.TabIndex = 5;
            this.picAddUser.TabStop = false;
            this.picAddUser.Click += new System.EventHandler(this.picAddUser_Click);
            this.picAddUser.MouseEnter += new System.EventHandler(this.SetStretchImage);
            this.picAddUser.MouseLeave += new System.EventHandler(this.SetDefaultImage);
            //
            // lblAddUser
            //
            this.lblAddUser.AutoSize = true;
            this.lblAddUser.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.lblAddUser.Location = new System.Drawing.Point(6, 112);
            this.lblAddUser.Name = "lblAddUser";
            this.lblAddUser.Size = new System.Drawing.Size(134, 20);
            this.lblAddUser.TabIndex = 6;
            this.lblAddUser.Text = "Adicionar Usuário";
            //
            // lblTimer
            //
            this.lblTimer.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Right)));
            this.lblTimer.AutoSize = true;
            this.lblTimer.Font = new System.Drawing.Font("Microsoft Sans Serif", 21.75F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.lblTimer.Location = new System.Drawing.Point(752, 22);
            this.lblTimer.Name = "lblTimer";
            this.lblTimer.Size = new System.Drawing.Size(92, 33);
            this.lblTimer.TabIndex = 8;
            this.lblTimer.Text = "00:00";
            //
            // pictureBox3
            //
            this.pictureBox3.Anchor =
((System.Windows.Forms.AnchorStyles)(((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Left)
            | System.Windows.Forms.AnchorStyles.Right)));
            this.pictureBox3.Enabled = false;
            this.pictureBox3.Image =
global::ACEEJ.WinForm.Properties.Resources.Appointment_Reminders_96;
            this.pictureBox3.Location = new System.Drawing.Point(485, 22);
            this.pictureBox3.Name = "pictureBox3";
            this.pictureBox3.Size = new System.Drawing.Size(133, 90);
            this.pictureBox3.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.CenterImage;
            this.pictureBox3.TabIndex = 7;
            this.pictureBox3.TabStop = false;
            //
```

```
            // tbcSchedules
            //
            this.tbcSchedules.Controls.Add(this.tpbMonday);
            this.tbcSchedules.Controls.Add(this.tpgTuesday);
            this.tbcSchedules.Controls.Add(this.tpgWednesday);
            this.tbcSchedules.Controls.Add(this.tpgThursday);
            this.tbcSchedules.Controls.Add(this.tpgFriday);
            this.tbcSchedules.Controls.Add(this.tpgSaturday);
            this.tbcSchedules.Dock = System.Windows.Forms.DockStyle.Fill;
            this.tbcSchedules.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.tbcSchedules.Location = new System.Drawing.Point(3, 153);
            this.tbcSchedules.Name = "tbcSchedules";
            this.tbcSchedules.SelectedIndex = 0;
            this.tbcSchedules.Size = new System.Drawing.Size(848, 369);
            this.tbcSchedules.TabIndex = 1;
            this.tbcSchedules.Tag = "";
            this.tbcSchedules.Selecting += new
System.Windows.Forms.TabControlCancelEventHandler(this.tbcSchedules_Selecting);
            //
            // tpbMonday
            //
            this.tpbMonday.Location = new System.Drawing.Point(4, 25);
            this.tpbMonday.Name = "tpbMonday";
            this.tpbMonday.Padding = new System.Windows.Forms.Padding(3);
            this.tpbMonday.Size = new System.Drawing.Size(840, 340);
            this.tpbMonday.TabIndex = 0;
            this.tpbMonday.Tag = "Monday";
            this.tpbMonday.Text = "Segunda-Feira";
            //
            // tpgTuesday
            //
            this.tpgTuesday.Location = new System.Drawing.Point(4, 25);
            this.tpgTuesday.Name = "tpgTuesday";
            this.tpgTuesday.Padding = new System.Windows.Forms.Padding(3);
            this.tpgTuesday.Size = new System.Drawing.Size(840, 340);
            this.tpgTuesday.TabIndex = 1;
            this.tpgTuesday.Tag = "Tuesday";
            this.tpgTuesday.Text = "Terça-Feira";
            this.tpgTuesday.UseVisualStyleBackColor = true;
            //
            // tpgWednesday
            //
            this.tpgWednesday.Location = new System.Drawing.Point(4, 25);
            this.tpgWednesday.Name = "tpgWednesday";
            this.tpgWednesday.Padding = new System.Windows.Forms.Padding(3);
            this.tpgWednesday.Size = new System.Drawing.Size(840, 340);
            this.tpgWednesday.TabIndex = 6;
            this.tpgWednesday.Tag = "Wednesday";
            this.tpgWednesday.Text = "Quarta-Feira";
```

```
//
// tpgThursday
//
this.tpgThursday.Location = new System.Drawing.Point(4, 25);
this.tpgThursday.Name = "tpgThursday";
this.tpgThursday.Padding = new System.Windows.Forms.Padding(3);
this.tpgThursday.Size = new System.Drawing.Size(840, 340);
this.tpgThursday.TabIndex = 3;
this.tpgThursday.Tag = "Thursday";
this.tpgThursday.Text = "Quinta-Feira";
this.tpgThursday.UseVisualStyleBackColor = true;
//
// tpgFriday
//
this.tpgFriday.Location = new System.Drawing.Point(4, 25);
this.tpgFriday.Name = "tpgFriday";
this.tpgFriday.Padding = new System.Windows.Forms.Padding(3);
this.tpgFriday.Size = new System.Drawing.Size(840, 340);
this.tpgFriday.TabIndex = 4;
this.tpgFriday.Tag = "Friday";
this.tpgFriday.Text = "Sexta-Feira";
//
// tpgSaturday
//
this.tpgSaturday.Location = new System.Drawing.Point(4, 25);
this.tpgSaturday.Name = "tpgSaturday";
this.tpgSaturday.Padding = new System.Windows.Forms.Padding(3);
this.tpgSaturday.Size = new System.Drawing.Size(840, 340);
this.tpgSaturday.TabIndex = 5;
this.tpgSaturday.Tag = "Saturday";
this.tpgSaturday.Text = "Sábado";
this.tpgSaturday.UseVisualStyleBackColor = true;
//
// timerClock
//
this.timerClock.Enabled = true;
this.timerClock.Interval = 1000;
this.timerClock.Tick += new System.EventHandler(this.timerClock_Tick);
//
// HandleControl
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.Controls.Add(this.tableLayoutPanel1);
this.Name = "HandleControl";
this.Size = new System.Drawing.Size(854, 525);
this.Load += new System.EventHandler(this.HandleControl_Load);
this.tableLayoutPanel1.ResumeLayout(false);
this.panel1.ResumeLayout(false);
this.panel1.PerformLayout();
```

```
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();
            ((System.ComponentModel.ISupportInitialize)(this.picSend)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.picConfig)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.picAddUser)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox3)).EndInit();
            this.tbcSchedules.ResumeLayout(false);
            this.ResumeLayout(false);

        }

        #endregion

        private System.Windows.Forms.TableLayoutPanel tableLayoutPanel1;
        private System.Windows.Forms.Panel panel1;
        private System.Windows.Forms.Label lblAddUser;
        private System.Windows.Forms.PictureBox picAddUser;
        private System.Windows.Forms.PictureBox pictureBox3;
        private System.Windows.Forms.TabControl tbcSchedules;
        private System.Windows.Forms.TabPage tpbMonday;
        private System.Windows.Forms.TabPage tpgTuesday;
        private System.Windows.Forms.TabPage tpgThursday;
        private System.Windows.Forms.TabPage tpgFriday;
        private System.Windows.Forms.TabPage tpgSaturday;
        private System.Windows.Forms.Label lblTimer;
        private System.Windows.Forms.TabPage tpgWednesday;
        private System.Windows.Forms.PictureBox picConfig;
        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label lblTextNextShot;
        private System.Windows.Forms.Label lblNextShot;
        private System.Windows.Forms.Timer timerClock;
        private System.Windows.Forms.PictureBox picSend;
        private System.Windows.Forms.Label label1;
    }
}
```