

# DESENVOLVIMENTO DE UM PROTÓTIPO PARA AUTOMAÇÃO DE COBERTURAS COM SENSOR DE CHUVA UTILIZANDO PLATAFORMA MICROCONTROLADORA ARDUINO

Renato Hildebrando Parreira<sup>1</sup>, Grazielle Borges de Souza<sup>2</sup>, Juan José dos Santos Souza<sup>3</sup>, Marcelo Nogueira<sup>4</sup>

**Abstract** — For the protection of sites discovered, are deployed awnings and roofs with manual intervention, was designed in the automation of these coverages to be closed when necessary for the protection of sites discovered, are deployed awnings and roofs with manual intervention, targeting this situation was thought in the automation these coverings to be closed when necessary. This article aims to address the problem of roofing manuals without the presence of an actuator (person) can not be closed in case of extreme necessity, as well as the exposure of the soil that can not be soaked with rain. Here is described a solution to this problem using an Arduino microcontroller board, rain sensor and servo motor, which are used in real-time embedded software.

**Index Terms** — Rain Sensor, Automation, Arduino, Roofing

## INTRODUÇÃO

O objetivo deste artigo é apresentar um protótipo com conceito de automação, monitorando o ambiente externo de uma forma simples, através de um sensor e atuador instalado no protótipo, o mesmo consiste na automação de coberturas para comércios e residências utilizando sensor de chuva e a plataforma eletrônica Arduino, com o objetivo de proteger áreas em céu aberto, caso, seja necessário, vigilância no local. Sempre que chover, automaticamente a cobertura será acionada e o local será coberto sem necessidade de intervenção humana, mantendo o local protegido e, após a chuva, a cobertura é automaticamente aberta expondo o local que fora escolhido para proteção.

Na seção 2 é apresentada a fundamentação teórica do trabalho. Na Seção 3 é apresentada a metodologia de desenvolvimento, e na Seção 4, as considerações finais.

## REFERENCIAL TEORICO

Nesta seção serão abordados os principais conceitos utilizados no desenvolvimento do protótipo de cobertura

automatizada para residências e comércios com sensor de chuva e plataforma microcontroladora Arduino.

## METODOLOGIA DE DESENVOLVIMENTO

O RUP (Rational Unified Process) é um framework genérico para processos de desenvolvimento de software, criado pela empresa Rational Software Corporation, que está fortemente centrado na arquitetura, funcionalidade (caso de uso) e o desenvolvimento iterativo e incremental (inspirado no ciclo de vida espiral de Boehm), que aplica a UML, para o projeto e a documentação. [4]

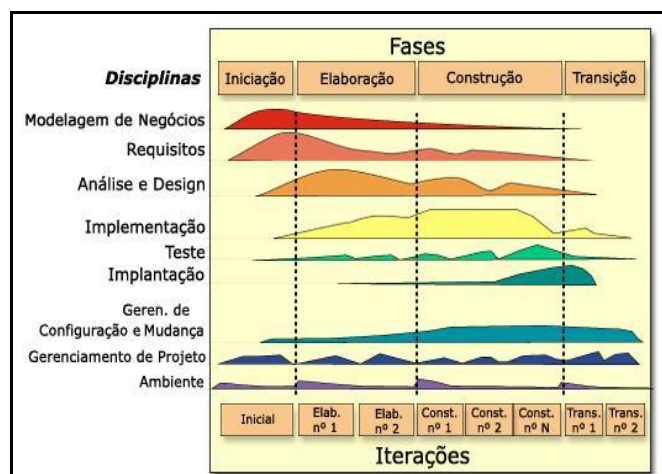


FIGURA 1

FASES DO RATIONAL UNIFIED PROCESS FONTE: (NOGUEIRA, 2009)

Para a elaboração do protótipo foram utilizados os seguintes documentos:

- Visão do Negócio
- Especificação de caso de uso de negócios
- Visão
- Especificação dos requisitos de software
- Documentos de Arquitetura de software
- Especificação de caso de uso de negócios

<sup>1</sup>Renato Hildebrando Parreira, Grupo de Pesquisa de Engenharia de Software aplicada à criação de Sistemas Críticos – UNIP, Rua Antonio Macedo, Parque São Jorge São Paulo – SP, Brazil, [renatodoryusen@hotmail.com.br](mailto:renatodoryusen@hotmail.com.br)

<sup>2</sup>Grazielle Borges de Souza, Grupo de Pesquisa de Engenharia de Software aplicada à criação de Sistemas Críticos – UNIP, Rua Antonio Macedo, Parque São Jorge São Paulo – SP, Brazil, [grazielleborgesdesouza@gmail.com](mailto:grazielleborgesdesouza@gmail.com)

<sup>3</sup>Juan José dos Santos Souza, Grupo de Pesquisa de Engenharia de Software aplicada à criação de Sistemas Críticos – UNIP, Rua Antonio Macedo, Parque São Jorge São Paulo – SP, Brazil, [juan\\_j16@hotmail.com](mailto:juan_j16@hotmail.com)

<sup>4</sup>Marcelo Nogueira Muniz - Universidade Paulista – UNIP, Rua Antônio de Macedo, 505, 03.087-040, Parque São Jorge – São Paulo - SP, Brazil, [marcelo@noginfo.com](mailto:marcelo@noginfo.com)

- Especificação de realização de casos de uso
- Plano de integração do build
- Plano de teste
- Sumário de Avaliação de testes
- Plano de implantação
- Plano de gerenciamento de configuração
- Plano de iteração
- Plano de métricas
- Plano de gerenciamento de risco
- Plano de desenvolvimento de software
- Glossário

## SOFTWARE EMBARCADO EM TEMPO REAL

Software embarcado é o software que é embutido em um equipamento eletrônico com microcontroladores, capaz de atuar de forma automática sem precisar da atuação de uma pessoa. Pode ser usado de exemplo para compreensão o sistema de injeção eletrônica de um automóvel, que permite a injeção de quantidade certa de combustível no motor proporcionando maior performance e economia de combustível. [3] O software embarcado não é visualizado pelo usuário, apenas a ação do objeto será percebida.

Segundo Shaw [6], são utilizados microprocessadores pouco conhecidos dos desenvolvedores e sistemas operacionais específicos, além de implementar conceitos de programação em tempo real (responder de imediato a eventos externos) e em muitos casos necessidades de segurança e confiabilidade bem acima das exigidas pelos sistemas comerciais. Devido a complexidade da arquitetura, pois envolve diversos componentes de hardware e de software e uma grande variedade de alternativas de soluções, que atendem aos requisitos do projeto, como nível de desempenho, consumo de potência e área ocupada, é necessária a adoção de metodologias que comecem o processo com níveis de abstração elevados, bem como sejam utilizadas ferramentas que automatizem ao máximo todas as etapas da mitologia.

O software de um sistema embarcado é diferente de um software de aplicação (software executado em um microcomputador) devido, principalmente, á forte interconexão com os dispositivos de hardware periféricos, as restrições impostos pela aplicação na qual ele esta inserido. [5]

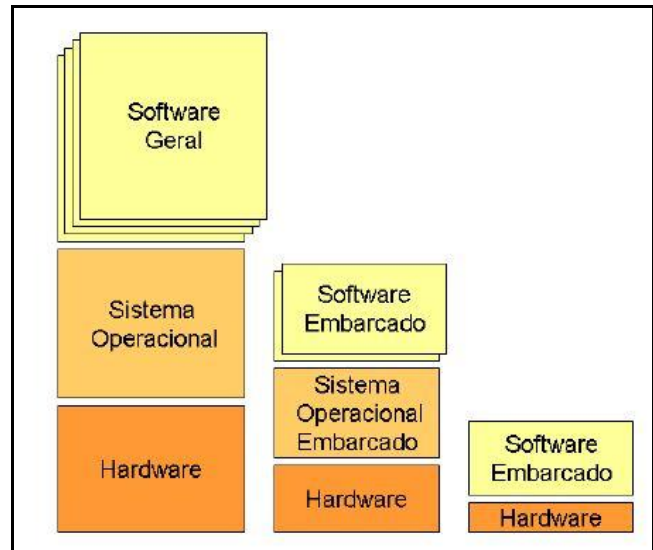


FIGURA 2

DIFERENÇA ENTRE SOFTWARE EMBARCADO E SOFTWARE EM GERAL

Sistema de tempo real são sistemas de computação que monitoram, respondem ou controlam um ambiente externo. Esse ambiente está conectado ao sistema de computação através de sensores, atuadores, e outras interfaces de entrada-saída. Podem constituir-se de objetos físicos ou biológicos de qualquer forma e estrutura. [7]

Na Figura 2 pode ser visto o processo de controle de sensor/atuador que mostra a lógica utilizada no processo de software embarcado em tempo real.

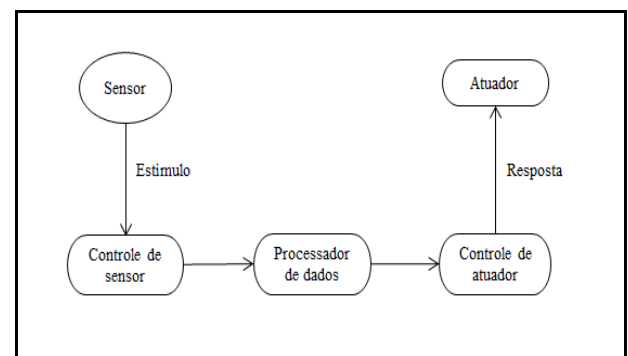


FIGURE 3

PROCESSO DE CONTROLE DE SENSOR-ATUADOR FONTE: ELABORADO PELOS AUTORES

## PLATAFORMA MICROCONTROLADORA ARDUINO

As placas de desenvolvimento baseadas em microcontroladores podem ser utilizadas em projetos de diversas áreas de conhecimento. O microcontrolador pode ser entendido como a incorporação de um microprocessador, e de sistemas de temporização, de aquisição e de comunicação em um mesmo circuito integrado em um exemplo de plataforma de desenvolvimento, baseado em

microcontrolador, largamente utilizado atualmente é o Arduino.[2]

Com o ambiente Arduino você pode construir objetos que possam responder e/ou controlar luz, som, toque e movimento. Arduino também é usada para criar uma incrível variedade de coisas, incluindo instrumentos musicais, robôs, esculturas de luz, jogos, móveis interativos e até roupas interativas. Arduino é mais conhecida pelo seu hardware, mas você também precisa do software para programar o hardware. Os dois, o hardware e o software são chamados “Arduino”. A combinação permite que você crie projetos nesse sentido e controle o mundo físico. [1]

A placa do Arduino é um circuito impresso que é especialmente projetado para utilizar um chip microcontrolador, como outros componentes de entrada e saída. Ele também possui vários outros componentes eletrônicos que são necessários para o microcontrolador funcionar ou para estender suas capacidades. [7]

O Arduino pode perceber o ambiente recebendo entradas de uma variedade de sensores e pode interagir com seus arredores controlando luzes, motores e outros atuadores. O microcontrolador na placa é programado utilizando a linguagem de programação Arduino (Baseada em conexões) e o ambiente de desenvolvimento Arduino (baseado em processamento). Projetos Arduino podem ser stand-alone (isolados) ou podem se comunicar com softwares rodando em um computador. [8]

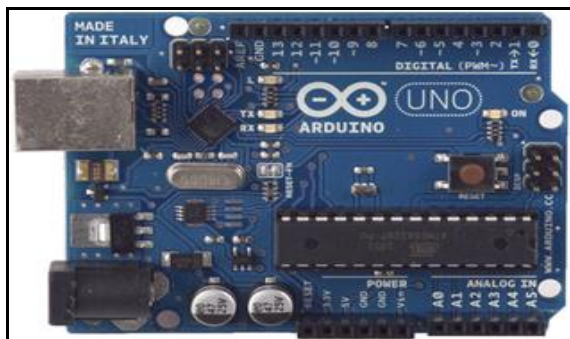


FIGURE 4

PLACA CONTROLADORA ARDUINO UNO

## MÉTODOS E MATERIAIS

Na primeira fase do projeto foi realizado o planejamento do protótipo, tendo como atividade principal a simulação do ambiente com o protótipo, componentes e materiais a serem utilizados, tempo gasto e a estimativa do custo.

O protótipo que simula uma cobertura automatizada, foi construído com madeira e folhas de alumínio, foi adicionado um sensor e um atuador para realizar a automação da cobertura além de uma fonte de tensão para alimentar a plataforma Arduino.

Na Figura 4 é apresentado o diagrama de caso de uso, auxiliar do planejamento e desenvolvimento do protótipo do projeto.

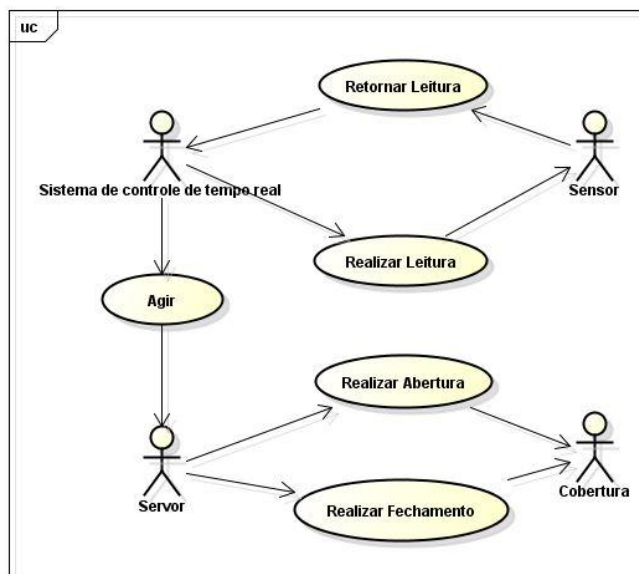


FIGURE 1

UML DO PROTÓTIPO FONTE: ELABORADO PELOS AUTORES

## PROTÓTIPO ELABORADO

O protótipo utilizado no projeto foi desenvolvido para simular uma cobertura residencial e/ou comercial, foram construídos dois cenários diferentes de cobertura. O material utilizado foi madeira e folhas de alumínio, também foi necessário a utilização de uma roldana para o movimento da cobertura. Para realizar a automação foi desenvolvido um sensor de chuva e implementado na plataforma Arduino, com um servo motor que é o responsável pela abertura e fechamento da cobertura. O software foi desenvolvido utilizando a linguagem de programação Arduino.

Nas figuras 5 e 6 podem ser vistos os modelos do protótipo utilizado para demonstração do ambiente descrito no protótipo.



FIGURE 5

CENÁRIO 1



FIGURE 6  
CENÁRIO 2

### CIRCUITO DA PLATAFORMA ARDUINO

Na Figura 7 é apresentado o desenho do protótipo, e na Figura 8 o circuito eletrônico elaborado, que representa o sistema de controle do protótipo em uma lógica sequencial. Para a elaboração do circuito foi utilizado:

- Um Servo motor (para atuar sobre a cobertura);
- Uma placa de circuito impresso com trilhas invertidas (utilizado como sensor);
- Uma placa microcontroladora Arduino;

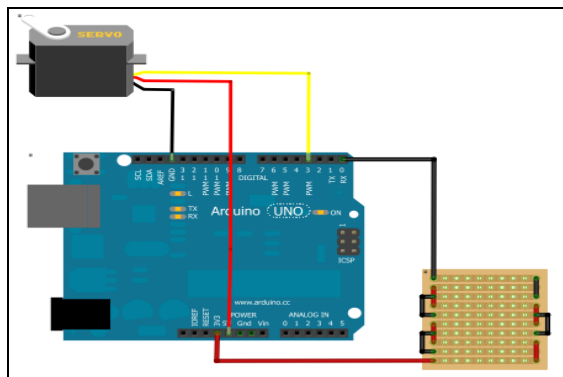


FIGURE 7  
MODELO PROTÓTIPO

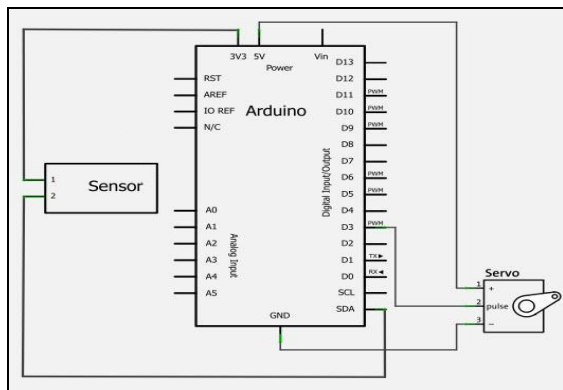


FIGURE 8  
ESQUEMATICO

### SENSOR DE CHUVA

O sensor de chuva foi desenvolvido pelo grupo como é ilustrado na Figura F, que consiste em uma placa de circuito inverso com trilhas ilustrado na Figura E. Utilizando um resistor de 10k com uma corrente de +5V. Em contato com a chuva o circuito do sensor fecha, passando a informação para o Arduino que irá interpretar qual será a ação (abrir ou fechar) de acordo com a regra do software embarcado na plataforma.

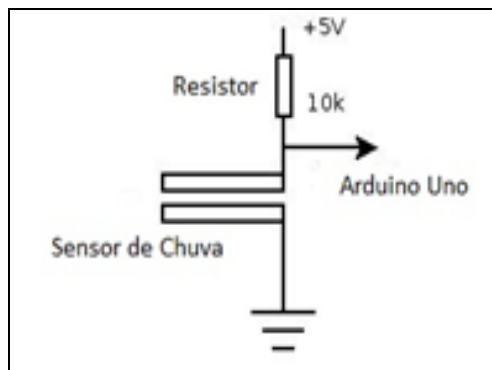


FIGURE 9  
SENSOR DE CHUVA ESQUEMÁTICO

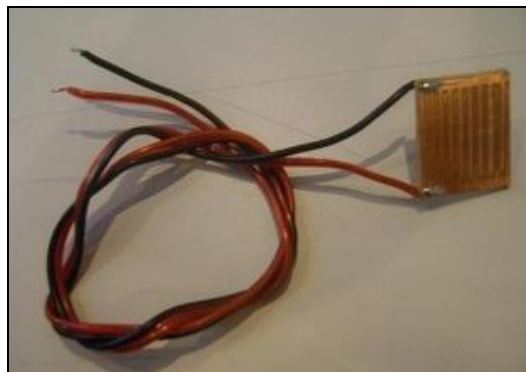


FIGURE 10  
SENSOR ELABORADO

### ALGORITMO IMPLEMENTADO

A lógica computacional elaborada foi realizada através da IDE do próprio Arduino. A lógica implementada (Figura 11), seguindo os seguintes passos:

- O microcontrolador inicia o sistema, as variáveis e as constantes definidas.
- Em seguida é realizada a leitura do sensor, a partir da comunicação entre o sensor e o microcontrolador.



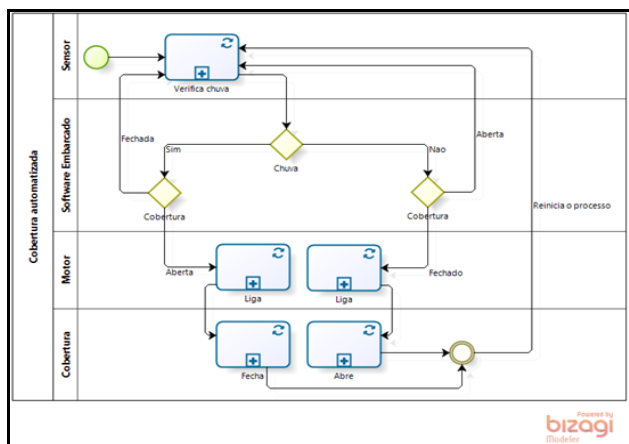


FIGURE 11

MODELO DE NEGOCIO DO PROTÓTIPO

## CONSIDERAÇÕES FINAIS

No decorrer do desenvolvimento do projeto foi criado dois protótipos de cobertura automatizada com sensor de chuva utilizando software embarcado em tempo real. Os protótipos foram desenvolvidos na intenção de simular o ambiente real, que ao ser implementado em um ambiente aberto de uma residência ou comércio, o sensor de chuva ficará exposto ao tempo e ao entrar em contato com a chuva a cobertura será automaticamente ativada e se fechará para a proteção do local, quando não for percebida chuva no ambiente e o sensor estiver seco, a cobertura será desativada e fechada automaticamente. Todo o processo é efetuado sem necessidade de intervenção manual.

É possível adaptar o projeto para outros tipos de cenário de cobertura, utilizando outros materiais. No futuro será desenvolvido outras etapas, como a inclusão do projeto no mercado e desenvolvimento do cenário no mundo real.

## REFERÊNCIAS

- [1] ARTHUR, W. Brian. The nature of technology: What it is and how it evolves. New York: Free Press, 2009.
- [2] FONSECA, E. G.P.; VEJA, A. S. Tutorial sobre Introdução a Projetos Utilizando o kit de Desenvolvimento Arduino. Anais XXXIX Congresso Brasileiro de Educação em Engenharia, Cobenge. Blumenau: FURB, 2011
- [3] STANKOVIC, John A. Misconceptions About Real-Time Computing. IEEE Computer: out. 1988.
- [4] NOGUEIRA, MARCELO. Engenharia de software. Um framework para a gestão de riscos em projetos de software. São Paulo: Ciência Moderna, 2009.
- [5] GOMES, HUMBERTO Metodologia de projeto de software embarcado voltada ao teste - Porto Alegre: Programa de Pós-Graduação em computação, 2010
- [6] SHAW, Alan C. Real-time Systems and Software. New York: John Wiley & Sons, 1. ed. 2001.
- [7] ALAN C. SHAW. Sistemas e Software de Tempo Real, Ed. John Wiley & Sons, 2001 NUSSEY, JOHN, Arduino for dummies, Wiley, 2013

- [8] ARDUINO. Disponível em <<http://arduino.cc>>. Acesso em maio de 2013.