CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA

SOUZA

ETEC ANTÔNIO DE PÁDUA CARDOSO

HABILITAÇÃO PROFISSIONAL TÉCNICA DE NÍVEL MÉDIO DE TÉCNICO EM INFORMÁTICA INTEGRADO AO ENSINO MÉDIO

**Adrielle Souza Sigismundo**

**Marcio Richard Catani Godoy**

**Maria Angélica Silva de Oliveira**

**Wesley Henrique Tardivo**

**X-SKULL**

**BATATAIS – SP**

**2015**

**Adrielle Souza Sigismundo**

**Marcio Richard Catani Godoy**

**Maria Angélica Silva de Oliveira**

**Wesley Henrique Tardivo**

**X-SKULL**

Trabalho de Conclusão de Curso apresentado ao

Curso Técnico em Informática da ETEC Antônio de Pádua Cardoso orientado pelos professores Luis Fernando da Silva e Valnei Henrique dos Reis como requisito parcial para obtenção do título de técnico em informática.

**BATATAIS – SP**

**2015**

*Agradecemos primeiramente aos nossos orientadores Professores Luis Fernando e Valnei Henrique, desprovida de palavras que possam fielmente retratar nossa gratidão, desejamos retribuir a competência, a sensibilidade e a disponibilidade com que sempre nos orientou, contribuindo para o aprimoramento intelectual, profissional e pessoal que acreditamos ter adquirido através do vínculo estabelecido.*

*“Às vezes, quando você inova, comete erros. É melhor admiti-los rapidamente continuar a melhorar suas outras inovações.”*

*- Steve Jobs*

**RESUMO**

Este trabalho apresenta como objetivo um modelo de crânio giratório, que se move de acordo com a direção(direita/esquerda) que detecta a presença de um corpo mais próximo. O chamamos de “X-Skull” ou “Crânio Extremo”, considerando a evolução do uso de servo motores, sensores ultrassônicos e plataforma arduino (Open source) o projeto mostra como funciona esta tecnologia, abordando as principais características com a programação C++ juntamente com a automação de objetos.

A proposta que este modelo oferece é a diversão e entretenimento das pessoas, destacando assim seus principais benefícios. O projeto poderá ser adaptado para ser utilizado em eventos e datas comemorativas, como natal, páscoa, halloween, etc.

Palavras-Chaves: Crânio giratório. Automação de objetos. Programação C++. Entretenimento.

**ABSTRACT**

As an aim this work has a rotating skull model that we call “X-Skull”, or “Extreme Skull”. Considering the evolution of the use of servo drives, of ultrasonic sensors, and of the Arduino platform (Open Source), the project shows how this technology works, mentioning the main characteristics with the C++ programming with the object automation.

The proposal that this model offers is the fun and the entertainment of people, thus emphasizing its main benefits. The project can be adapted to be used at events and commemorative dates, such as Christmas, Easter, Halloween, and others.

LISTA DE FIGURAS

**Figura 1 - Crânio..............................................................................................10**

**Figura 2 – Sensor Ultrassônico......................................................................11**

**Figura 3 – Protoboard.....................................................................................12**

**Figura 4 – Servo Motor....................................................................................13**

**Figura 5 e 6 – Placa Arduino...........................................................................14**

**Figura 7 – Jumpers…………………………………………………………………18**

SUMÁRIO

[INTRODUÇÃO](#_Toc437401727) 9

[1 DESENVOLVIMENTO 1](#_Toc437401728)0

[1.1 Sobre os Materiais Utilizados 1](#_Toc437401729)0

[1.1.2 Crânio 1](#_Toc437401730)0

1.1.[3 Sensor de Presença 1](#_Toc437401731)1

[1.1.4 Protoboard 1](#_Toc437401732)1

[1.1.5 Servo Motor 1](#_Toc437401733)2

[1.1.6 Arduino](#_Toc437401734) 13

[1.1.7 Hardware e Software da placa](#_Toc437401735) 14

[**1.1.8 Bibliotecas..................................................................................................**](#_Toc437401736).**15**

[**1.1.9 Estruturas e Funções.................................................................................**](#_Toc437401737) **15**

[1.1.10 Jumper](#_Toc437401738) 18

[2 CONCLUSÃO](#_Toc437401739) 19

[REFERÊNCIA](#_Toc437401740) 20

INTRODUÇÃO

O presente trabalho é chamado de “X-Skull”, mais concretamente é um crânio giratório que se move de acordo com a presença detectada nos sensores. Foi realizado a partir de debates, fóruns na internet, troca de ideias com os orientadores e várias pesquisas.

Para o resultado final ter sido realizado com sucesso, realizou-se vários testes e o primeiro protótipo foi criado: o crânio de Halloween, que trouxe resultados positivos. Então a partir daí o projeto foi concretizado com a ideia de usar a caveira, e apresentado produtivamente.

O objetivo principal é o entretenimento e interação do público com o projeto, por ser de fácil manuseio e o crânio adaptável, ele pode ser trocado em festas de aniversário por exemplo, com o crânio de acordo com o tema da festa. Acredita-se em unanimidade entre os criadores que o “X-Skull” ganhará espaço no mercado por ser interativo e atrativo.

# 1 DESENVOLVIMENTO

Para o desenvolvimento deste projeto, foram necessários os seguintes materiais:

* Arduino Uno;
* Sensor de Presença;
* Servo Motor;
* Jumpers;
* Crânio Plástico.

**1.1 Sobre os materiais utilizados**

Nos capítulos seguintes, há explicações sobre cada item, já que, para a realização do projeto, foi necessário um conhecimento maior sobre estes; para que fossem utilizados corretamente e que gerassem o resultado desejado.

**1.1.2 Crânio**

No projeto em questão foi decidido que seria usado uma caveira para que causasse uma certa surpresa no público que a visse funcionando. Realizou-se várias pesquisas em sites para que houvesse uma melhoria na ideia, e que tivesse além de um preço acessível, também um peso ideal para o servo motor o movimentar, portanto foi usado como material o plástico como mostra a figura acima por ser leve e de fácil manuseio.



Figura 1 – Crânio – Acervo Pessoal

**1.1.3 Sensor Ultrassônico de Movimento e Presença – HC SR04**

Esses sensores se caracterizam por operar por um tipo de radiação não sujeita a interferência eletromagnética e totalmente limpa, o que pode ser muito importante para determinados tipos de aplicações.

Podendo operar de modo eficiente detectando objetos em distâncias que variam entre milímetros até vários metros, eles podem ser empregados para detectar os mais variados tipos de objetos e substâncias.

O comprimento de onda usado e, portanto, as frequências são muito importantes nesse tipo de sensor, pois ele determina as dimensões mínimas do objeto que pode ser detectado, só ocorre reflexão em intensidade suficiente para se obtiver um bom sinal, quando o objeto tem dimensões que se aproximam do comprimento de onda do sinal. Os sensores são capazes de executar tarefas simples de detecção de forma mais precisa e eficiente.

Esses sensores podem ser utilizados nas indústrias para verificar a presença de peças em uma linha de montagem, para medir o tamanho destas peças e para verificar o estado dos produtos, ou podem ser usados para vários fins. Os sensores são capazes de executar tarefas simples de detecção de forma mais precisa e eficiente.

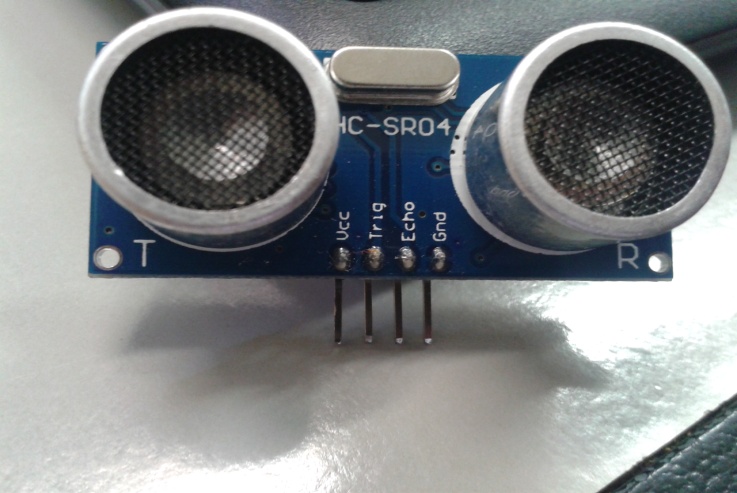


Figura 2 – Sensor Ultrassônico – Acervo Pessoal

**1.1.4 Protoboard**

O Protoboard é uma ferramenta para criação rápida de protótipos e circuitos eletrônicos sem a necessidade de soldar componentes. Nele, são inseridos os terminais de um componente nos furos. Assim, os terminais serão conectados eletricamente a outros furos. O importante é saber quais são estes outros furos que o terminal estará conectado.

No centro do protoboard há linhas marcadas com números e dez colunas marcadas com letras. As colunas laterais são geralmente utilizadas como linhas de alimentação elétrica e qualquer furo internamente conectado a todos os outros furos da mesma coluna. Isto é muito útil quando o seu projeto utiliza mais de uma fonte de alimentação (5V e 3V por exemplo). Na maioria dos casos é importante ligar os dois GNDs, ficando com uma única referência.

Os outros furos, são as linhas do centro. Estas linhas são conectadas horizontalmente. Se utilizar o sistema de coordenadas de letras e números, o furo do canto superior esquerdo será o furo “a1″. O furo “a1″ se conecta a b1, c1, d1 e e1”. Existe um vão ao centro separando as linhas, e elas nãose conectam através deste vão. Existe um vão bem no meio da protoboard, a razão para esta separação ao centro é que ela serve perfeitamente para se colocar circuitos integrados (CIs ou chips) do tipo DIP (Dual Inline Package) sobre este vão, permitindo acesso a todos os pinos do circuito integrado.

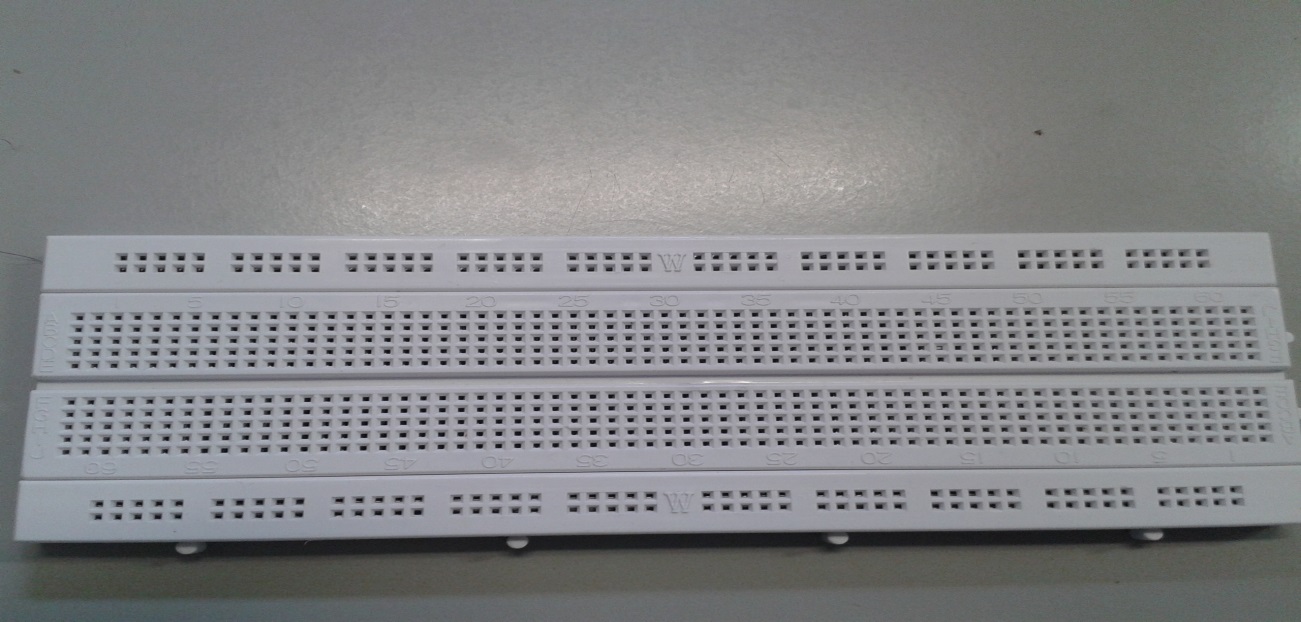
****

Figura 3 - Protoboard – Acervo Pessoal

**1.1.5 - Servo Motor**

Um servo motor é um atuador eletromecânico utilizado para posicionar e manter um objeto em uma determinada posição. Para isso, ele conta com um [controle de malha fechada](http://pt.wikipedia.org/wiki/Controle_de_malha_fechada) que permite o cálculo do erro de sua posição atual de modo à sempre se manter na posição desejada, eles são empregados no controle de pequenos robôs ou em projetos de menores proporções ou em sistemas de automação.

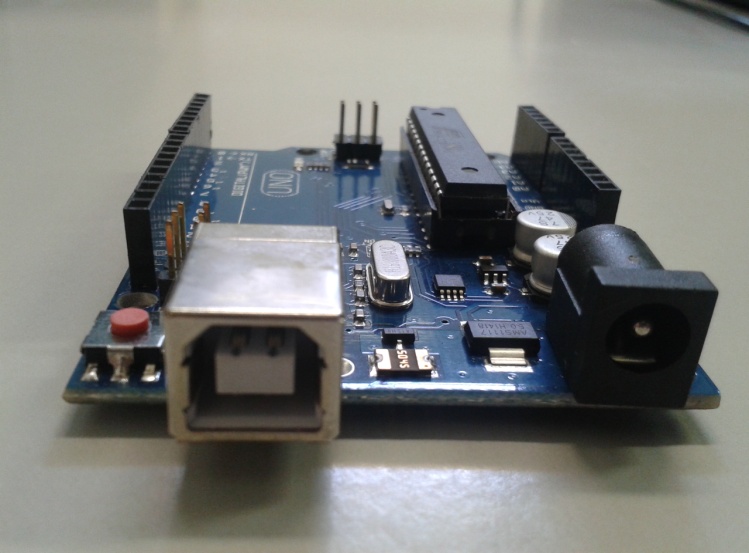
Ele é alimentado com tensões de 5 V e recebe um sinal no formato PWM (Pulse Width Modulation). Este sinal é 0 V ou 5 V. O circuito de controle do servo fica monitorando este sinal em intervalos de 20 ms. Se neste intervalo de tempo, o controle detecta uma alteração do sinal na largura do sinal, ele altera a posição do eixo para que a sua posição coincida com o sinal recebido.

No projeto “X-Skull”, para controlar o servo motor foi usado o arduino que com sua programação embutida quando este fizesse a leitura ele se posicionaria conforme o movimento de um objeto próximo aos sensores usados no projeto.

****

Figura 4 - Servo Motor – Acervo pessoal

**1.1.6 Arduino**

Arduino é uma plataforma deprototipagemeletrônica de hardware livre e que tem circuitos que são usadas para entrada e saída de dados servindo para controlar motores e o que for necessário. Plataformas como o arduino aumentaram muito o acesso a esses dispositivos uma vez que abstraiu várias camadas do desenvolvimento, ou seja, foi feita uma padronização do hardware e software básico para os microcontroladores e o programador pode atuar com funções que já foram escritas. A plataforma é composta essencialmente de duas partes: O Hardware e o ****Software.

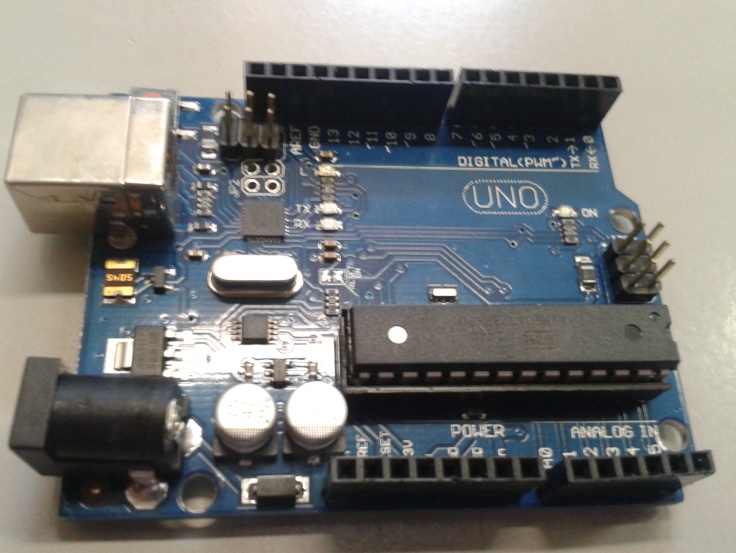
****

Figura 5 e 6 – Placa Arduino – Acervo Pessoal

**1.1.7 – Hardware e Software da placa**

O hardware do arduino permite usar a placa com diversas fontes de energia, baterias e fontes de alimentação e nele próprio dá para colocar circuitos de conexão.

O Arduino é um compilador gcc (C e C++) baseado em Wiring e que usa uma interface gráfica construída em Java baseado no projeto Processing. Tudo isso se resume a um programa IDE (ambiente de desenvolvimento integrado) muito simples de usar e de estender com bibliotecas que podem ser facilmente encontradas na internet.

Depois de criar o programa e compilar usando a IDE, o código gerado é enviado para a placa onde é gravado dentro do chip.

As funções da IDE do Arduino são basicamente duas: Permitir o desenvolvimento de um software e enviá-lo à placa para que possa ser executado.

**1.1.8 Bibliotecas**

Quando se faz tarefas mais complexas ou se utiliza algum outro circuito conectado ao Arduino, um recurso muito importante são as bibliotecas.

Uma biblioteca é basicamente composta de código fonte adicional que se adiciona ao projeto por meio do comando include**.** Uma biblioteca do Arduino se apresenta normalmente como uma ou mais classes que possuem funções, os métodos, para acionar dispositivos, configurá-los ou executar alguma outra tarefas

**1.1.9 Estruturas e Funções**

**Void**

A palavra-chave void é usada apenas em declarações de funções. Ela indica que a função não deve enviar nenhuma informação de retorno à função que a chamou.

**Exemplo:**

// ações são realizadas nas funções "setup" e "loop"  
// mas nenhuma informação é enviada ao programa mais amplo  
  
void setup()  
{  
 // ...  
}  
  
void loop()  
{  
 // ...  
}

**For**

A sentença for é utilizada para repetir um bloco de código delimitado por chaves. Um contador com incremento normalmente é usado para controlar e finalizar o loop. A sentença for é útil para qualquer operação repetitiva, e é frequentemente usada com arrays para operar em conjuntos de dados ou de pinos.

Há três partes no cabeçalho do for:

for (inicialização; condição; incremento) {

//sentença(s);

}

A inicialização ocorre apenas uma vez. Cada vez que o circuito é executado a condição é verificada; se for verdadeira o bloco de código e o incremento é executado, e então a condição é testada novamente. Quando a condição se torna falsa o circuito termina.

**If (condicional)**

If, que é usado juntamente com um operador de comparação, verifica quando uma condição é satisfeita, como por exemplo um input acima de um determinado valor. O formato para uma verificação if é:

if (alguma variável > 50)  
{  
 // faça alguma coisa  
}

O programa checa se alguma variável (colocar acentos em nomes de variáveis não é uma boa ideia) é maior que 50. Se for, o programa realiza uma ação específica. Colocado de outra maneira se a sentença que está dentro dos parêntesis é verdadeira o código que está dentro das chaves roda; caso contrário o programa salta este bloco de código.

As chaves podem ser omitidas após uma sentença if se só houver uma única linha de código (definida pelo ponto e vírgula) que será executado de modo condicional:

if (x > 120) digitalWrite(LEDpin, HIGH);   
  
if (x > 120)  
digitalWrite(LEDpin, HIGH);   
  
if (x > 120) {digitalWrite(LEDpin, HIGH);} // todos são corretos

A sentença que está sendo verificada necessita o uso de pelo menos um dos operadores:

### Operadores de comparação:

**x == y** (x é igual a y)  
**x != y** (x é não igual a y)  
**x < y** (x é menor que y)   
**x > y** (x é maior que y)   
**x <= y** (x é menor ou igual a y)   
**x >= y** (x é maior ou igual a y)

**Delay (ms)**

**Descrição**

Suspende a execução do programa pelo tempo (em milissegundos) especificado como parâmetro. (Em um segundo há 1.000 milissegundos.)

**Parâmetros**

ms (unsigned long): o número de milissegundos em que o programa ficará com a execução em suspenso.

**Retorno**

Nenhum

**Exemplo**

int ledPin = 13; // LED conectado ao pino digital 13  
  
void setup()  
{  
 pinMode(ledPin, OUTPUT); // marca o pino digital como saída  
}  
  
void loop()  
{  
 digitalWrite(ledPin, HIGH); // acende o LED  
 delay(1000); // espera por um segundo  
 digitalWrite(ledPin, LOW); // apaga o LED  
 delay(1000); // espera por um segundo  
}

**While**

**Descrição**

While fará com que o bloco de código entre chaves se repita continua e indefinidamente até que a expressão entre parêntesis () se torne falsa. Algo tem que provocar uma mudança no valor da variável que está sendo verificada ou o código vai sempre ficar dando voltas dentro do while. Isto poderia ser o incremento de uma variável ou uma condição externa como o teste de um sensor.

**Sintaxe**

while(expressão){  
 // código  
}

**Parâmetros**

Expressão - uma sentença booleana em C que possa ser verificada como verdadeira ou falsa

**Exemplo**

var = 0;  
while(var < 200){  
 // algum código que se repete 200 vezes  
 var++;  
}

**1.1.9 Jumpers**

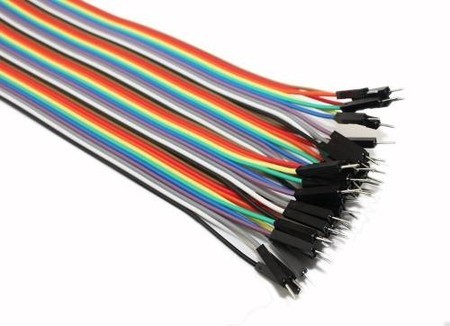


Figura 7 – Jumpers – Acervo Pessoal

Jumper é uma ligação móvel entre dois pontos de um circuito eletrônico. É, geralmente, uma pequena peça plástica isolante que contém uma peça metálica em seu interior, responsável pela condução de eletricidade. Em circuitos impressos (PCB em inglês) são responsáveis por desviar, ligar ou desligar o fluxo elétrico permitindo configurações específicas por meio físico do hardware em questão. Em discos rígidos (HD) eles configuram o modo como a placa-mãe reconhece o disco rígido no sistema. Podendo configurá-los como "master" ou "slave", ou mesmo pode limitá-los a 32GiB de capacidade.

# 2 CONCLUSÃO

# Através deste trabalho, chega-se a conclusão que a programação de equipamentos eletrônicos, ou automação de componentes, é uma tecnologia recente, que vem ganhando espaço nos últimos anos. Nota-se que esse tipo de recurso é uma tendência, tanto pra uso do entretenimento como comercial. Além disso, o uso da plataforma Arduino, por ser open source, possui baixo custo, sendo acessível para qualquer pessoa que possua noções de lógica de programação e conhecimento da linguagem de programação C++. Dessa forma, podemos afirmar que é possível transformar um pequeno projeto em algo comercial.

# REFERÊNCIAS

Como redigir a introdução de uma monografia. Disponível em:

<<http://pt.slideshare.net/BiblioEscolarOurique/como-redigir-a-introduo-e-a-concluso-de-um-trabalho-escrito-10140564> >. Acesso em 21 ago. 2015.

Esqueleto de monografia. Disponível em:

<<http://pt.slideshare.net/ksilvadacosta/tcc-exemplo-esqueleto>>. Acesso em 1 dez. 2015.

Modelo de referência. Disponível em:

<<http://www.ufrgs.br/bibenf/arquivos/EXEMPLOS%20DE%20REFERENCIAS.pdf>>. Acesso em 13 out. 2015.

Sensores Ultrassônicos. Disponível em:

<http://www.sabereletronica.com.br/artigos/1753-sensores-ultra-snicos > Acesso em 20 nov. 2015

Exemplos de monografias. Disponível em:

<<http://www.monografias.poli.ufrj.br/monografias/monopoli10001369.pdf>> Acesso em 20 nov. 2015

Descrição de arduino. Disponível em:

<http://www.robotizando.com.br/curso\_arduino\_o\_que\_e\_arduino\_pg1.php> Acesso em 20 nov. 2015

Descrição de arduino. Disponível em:

<http://robolivre.org/conteudo/arduino> Acesso em 16 nov. 2015

Servo Motores. Disponível em:

http://www.ebah.com.br/content/ABAAABaCoAI/trabalho-sobre-servomotor-controlador-pid > Acesso em 16 nov. 2015

Normas ABNT. Disponível em:

<http://viacarreira.com/regras-da-abnt-para-tcc-conheca-principais-normas> Acesso em 16 nov. 2015

.