



**INSTITUTO  
FEDERAL**

Santa Catarina

---

Câmpus  
São José

## **Relatório**

Cenário 2 - Travessia Controlada por Botoeira com Sinalização Noturna, Sinalização Piscante,  
Avisos Sonoros

**Rhenzo Hideki Silva Kajikawa**  
**Matheus Pires Salazar**

26 de Setembro de 2023

# Sumário

<b>1. Introdução .....</b>	<b>3</b>
1.1. Objetivo .....	3
1.2. Motivação .....	3
<b>2. Descrição do Projeto .....</b>	<b>4</b>
2.1. Cenário 2 - Descrição .....	4
2.2. Procedimento .....	4
2.3. Componentes utilizados .....	6
2.4. Sistema Completo .....	7
2.4.1. Elementos lógicos do sistema .....	7
2.4.2. RTL Viewer .....	8
<b>3. Implementação na placa .....</b>	<b>10</b>
3.1. Pinagem .....	10
3.2. Resultados .....	10
<b>4. Conclusão .....</b>	<b>11</b>

# **1. Introdução**

## **1.1. Objetivo**

Este projeto, feito na matéria de Dispositivos lógicos programáveis , tem como objetivo simular uma situação de transito , onde, como descrito, é feita a travessia de pedestres em diferentes cenários tanto de manhã quanto a noite . Além de tentar otimizar o transito , evitando fechamentos de semáforos da via principal de forma desnecessária.

## **1.2. Motivação**

Em aula foram ensinados novos conceitos de VHDL, e agora como foi aplicado desde o mais básico até conceitos mais complexos. Dessa forma esse projeto visa certificar que foram aprendidos todos esses conhecimentos.

## **2. Descrição do Projeto**

### **2.1. Cenário 2 - Descrição**

Visa garantir uma travessia de pedestres diurna e noturna segura e consciente. Ao acionar a botoeira, será ativada uma iluminação branca sobre a faixa de passagem zebraada e nas áreas de espera dos pedestres, assegurando melhor visibilidade e segurança para o pedestre a noite. Simultaneamente, o semáforo emitirá sinais visuais e sonoros, indicando ao pedestre que o botão foi acionado com sucesso e alertando motoristas sobre a intenção de travessia. Durante a fase de liberação para veículos, o semáforo do pedestre permanecerá vermelho, economizando energia até que o botão seja acionado. Após a solicitação, os grupos focais do pedestre exibirão luz verde em ambos os lados da via, enquanto o semáforo dos carros exibirá sinal vermelho, garantindo a máxima segurança para os pedestres e reforçando a prioridade de travessia.

Para orientar pedestres de maneira eficaz, o semáforo do pedestre apresentará um contador regressivo, indicando o tempo restante para a travessia. O tempo total de travessia será ajustável, permitindo personalização conforme as necessidades locais. Nos últimos 30% do tempo, o sinal verde do semáforo do pedestre piscará, visualmente alertando que o tempo para a travessia está se encerrando. É importante ressaltar que a iluminação estará ativa apenas durante o tempo em que a botoeira foi acionada até 5 segundos após o término do tempo de travessia. Este ajuste visa otimizar o consumo de energia e garantir que a iluminação cumpra sua função apenas quando necessária.

### **2.2. Procedimento**

Fora de aula foram decididas como seria a divisão do projeto. Foi decidido que era necessário ilustrar como seria feito a maquina de estados antes mesmo de começar o código em vhdl.

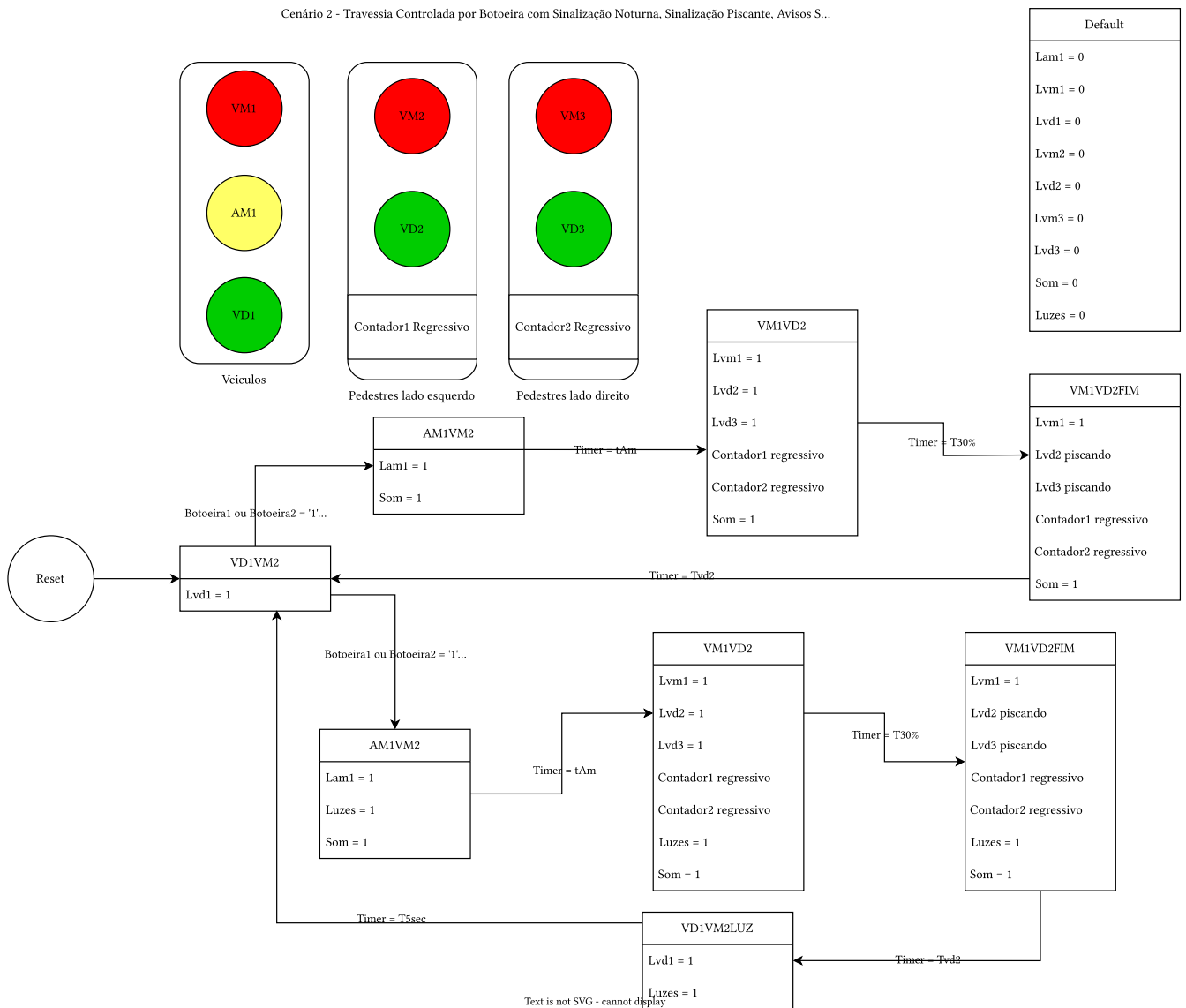


Figura 1: Elaboração das máquinas de estado

Fonte: Elaborada pelo autor

Nesse levantamento baseado nos semáforos visto em aula foi possível visualizar quais seriam as diferenças e semelhanças, além de ter ideia das entradas e saídas da máquina de estado do cenário 2.

Após pensar em como seria levantado a máquina de estado, fizemos uma varredura em quais componentes iriam ser necessários para a simular o cenário escolhido, e também atender as requisições feitas pelo professor.

Os componentes que foram escolhidos para compor o resto do projeto foram , um conversor bin2bcd , 2 conversores bcd2ssd e um divisor de clock.

A visualização do projeto ficou dessa forma:

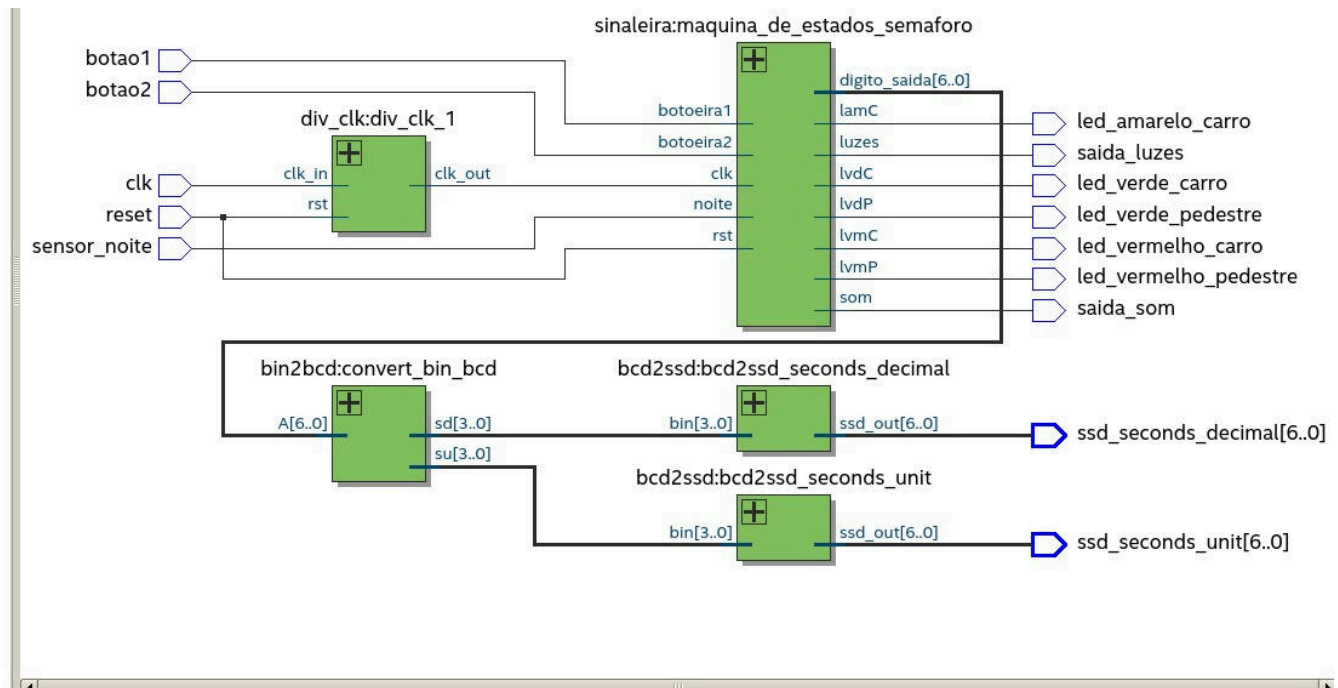


Figura 2: Elaboração das máquinas de estado  
Fonte: Elaborada pelo autor

### 2.3. Componentes utilizados

O componentes utilizados para o projeto podem ser visualizados na Figura 2.

No total foram utilizados 2 displays de sete segmentos , 2 conversor de BCD para ssd , 1 conversor de Binário para BCD , 1 maquina de estados , 1 divisor de clock e 1 clock de 50 MHz , 2 botões para pedestres , 2 LEDs para simbolizar som e luz , 3 LEDs para semáforo de carros e mais 2 LEDs para semáforo de pedestres.

Como dito anteriormente , nesse projeto foi abordado maquinas de estados , a partir disso também foi trabalhado com o uso de 4 componentes menores.

Um divisor de clock com 2 entradas , reset e clock in , e uma saída clock out . Esse divisor tem como objetivo diminuir os pulsos de enable dos segundos , podendo ajustar toda contagem do sistema para diferentes clocks . Apenas trocando o valor genérico do componente “div”. Esse divisor de clock foi utilizado anteriormente no projeto do relógio que foi feito em sala , dessa maneira para otimizar o tempo foi decidido reutiliza-lo.

Para a maquina de estado foram criada com 5 entradas , a divisão delas sendo , 1 entrada padrão de reset, 1 entrada para o clock onde é conectada com o divisor de clock. 2 entradas que são das botoeiras para os pedestres que quando apertarem faz o estado mudar para abrir o semáforo dos pedestres e por ultimo uma entrada do sensor de noite.

Existem 8 saídas para a maquina de estado , 3 LEDs que servem para o semáforo dos carros , 2 LEDs para o semáforo dos pedestres junto a 1 saída de contagem e uma saída de som.

A saída de contagem tem como objetivo passar por um conversor binário para bcd , desse saem 2 pontos , 1 deles é para o decimal e outro a unidade que passa para um conversor bcd2ssd , esses são mostrados em 2 display de 7 segmentos que é habilitado quando o pedestre deve passar.

## 2.4. Sistema Completo

Após ter todos os componentes do projeto foi feita a junção deles.

### 2.4.1. Elementos lógicos do sistema

Flow Status	Successful - Fri Dec 15 14:31:58 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Standard Edition
Revision Name	Cenario2
Top-level Entity Name	div_clk
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	11
Total registers	7
Total pins	3
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

Figura 3: RTL viewer do divisor de clock

Fonte: Elaborada pelo autor

Flow Status	Successful - Fri Dec 15 14:39:38 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Standard Edition
Revision Name	Cenario2
Top-level Entity Name	bin2bcd
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	98
Total registers	0
Total pins	15
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

Figura 4: RTL viewer do bin2bcd

Fonte: Elaborada pelo autor

Flow Status	Successful - Fri Dec 15 14:33:26 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Standard Edition
Revision Name	Cenario2
Top-level Entity Name	bcd2ssd
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	7
Total registers	0
Total pins	11
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

Figura 5: RTL viewer do bcd2ssd

Fonte: Elaborada pelo autor

## 2.4.2. RTL Viewer

Aqui estão os RTLS viewers para cada componente:

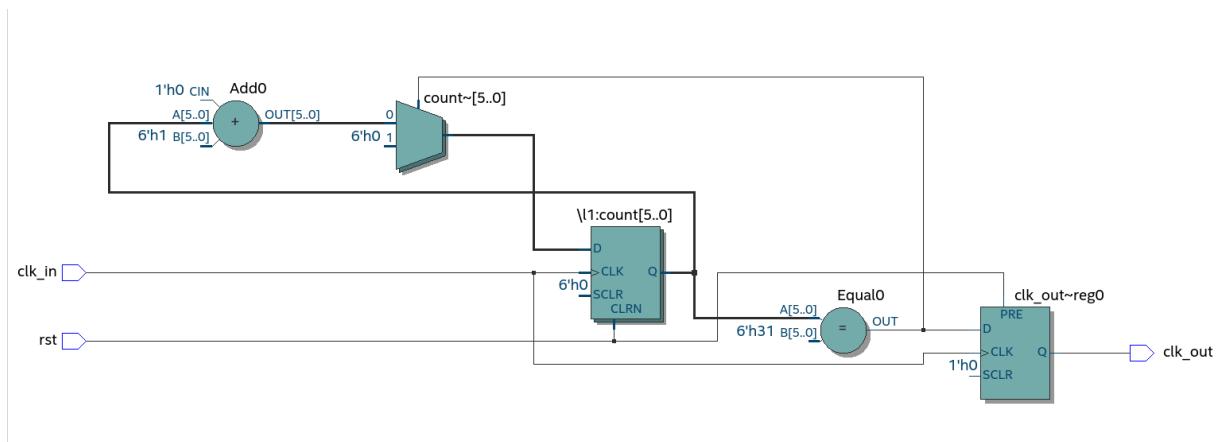


Figura 6: RTL viewer do divisor de clock

Fonte: Elaborada pelo autor



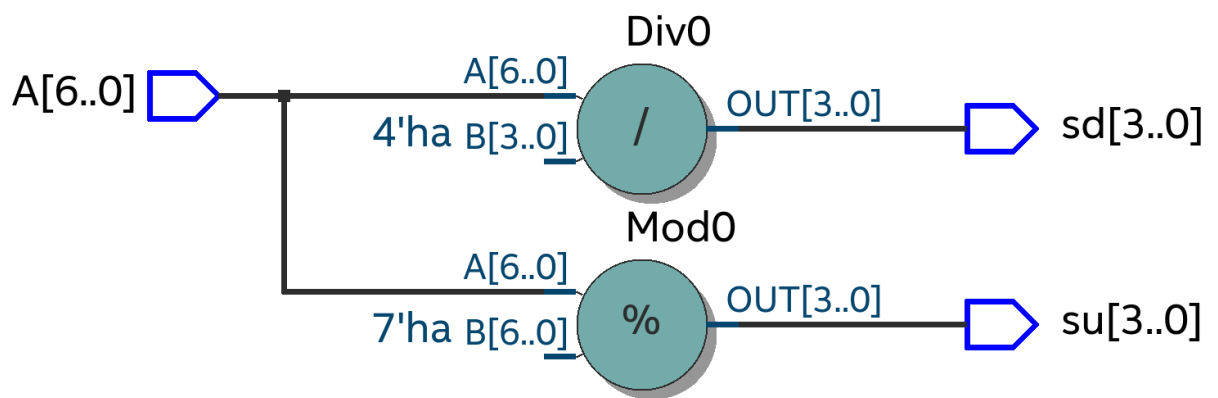


Figura 7: RTL viewer do bin2bcd  
 Fonte: Elaborada pelo autor

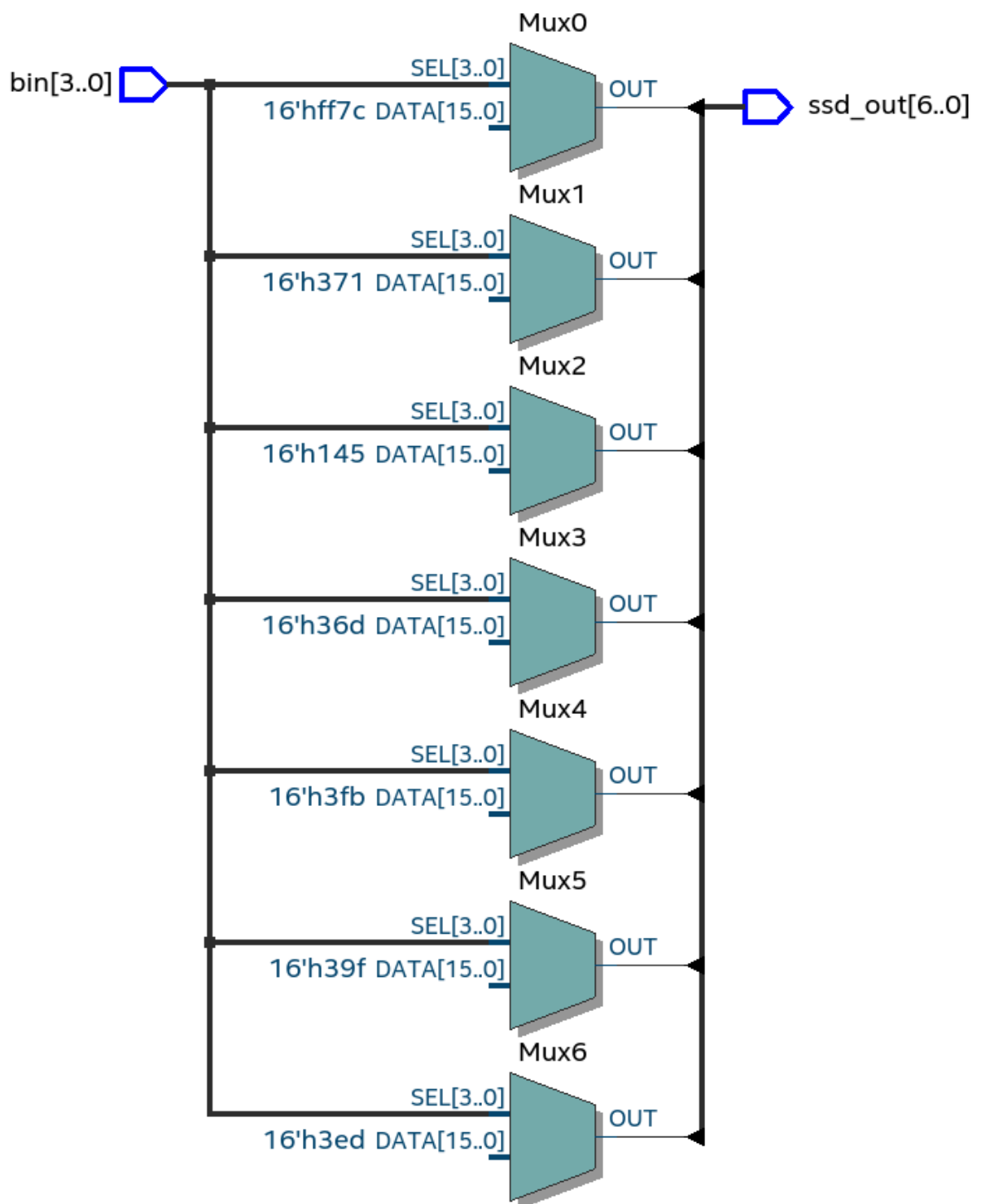


Figura 8: RTL viewer do bcd2ssd

Fonte: Elaborada pelo autor

### 3. Implementação na placa

#### 3.1. Pinagem

#### 3.2. Resultados

## **4. Conclusão**