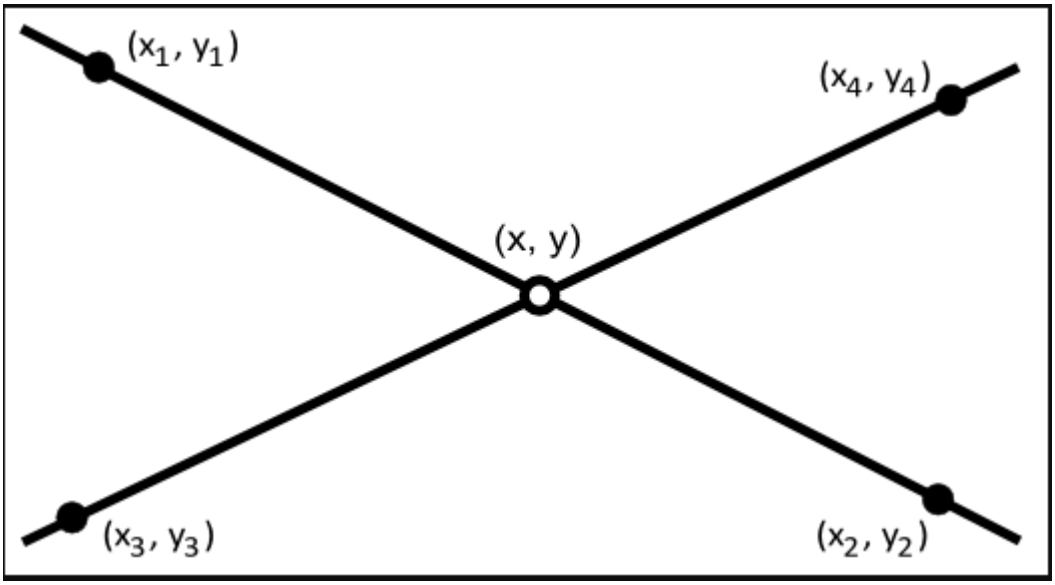
Tutorial do Projeto

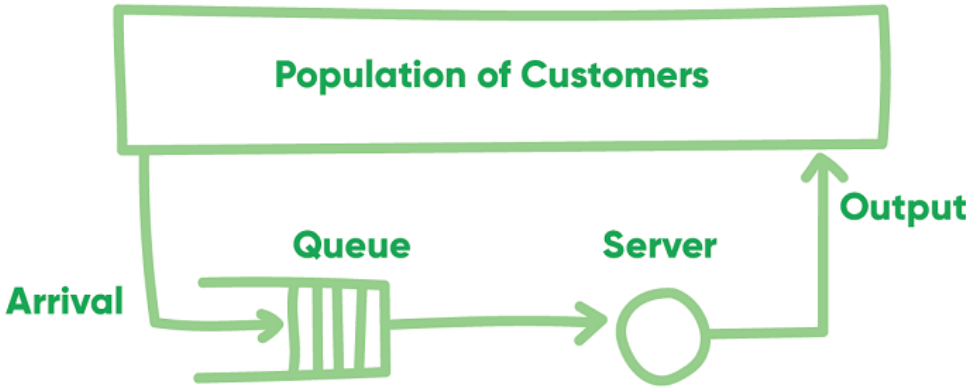
A ideia deste tutorial é explicar de maneira didática tudo que foi desenvolvido no projeto da disciplina de Projeto Integrador III do IFSC, afim de um futuro leitor deste repositório conseguir entender e executar este projeto, podendo ainda dar continuidade no projeto em uma situação futura. O projeto foi desenvolvido em linguagem C pela familiaridade com esta linguagem de programação, mas ele poderia ser desenvolvido em qualquer outra, a qual o leitor se sentir à vontade. No projeto foram desenvolvidos três modelos para um controle de tráfego de um cruzamento em X onde há apenas pista simples. Há uma Rua A e há uma Rua B.



O projeto procura modelar um cruzamento em X como este retratado na figura em questão. Na Rua A(x1,y1) veículos trafegam em pista simples até as coordenadas (x2,y2). Na Rua B(x3,y3) veículos trafegam em pista simples até as coordenadas (x4,y4). Repare que há um ponto de intersecção (ponto em comum) neste cruzamento. Neste ponto em comum há um controlador de trânsito(semáforo), o qual terá alguma política de abertura e fechamento para

veículos trafegarem. Este é um ponto muito importante para este projeto, qual política adotada em um controlador de tráfego urbano apresentará um melhor desempenho, se comparada a três políticas distintas? A ideia deste estudo é fazer comparações entre modelos distintos de acordo com a política adotada e verificar o seu comportamento com uma simulação computacional e extrair resultados passíveis de comparações entre eles.

Antes de iniciar o tutorial do projeto é preciso compreender alguns conceitos utilizados na criação dos modelos de estudo.



Imagine que esta imagem representa um sistema com uma entrada e uma saída. Ela representa a modelagem do problema estudado. Veículos entram neste sistema pelo Arrival e assim entram na fila (Queue), em uma fila é respeitada a seguinte regra First In, First out(FIFO), ou seja se um veículo ser o primeiro a entrar na Queue ele será o primeiro a sair da mesma. Em um sistema como este foi pensado na seguinte ideia para o Arrival, esta chega de veículos no sistema será representada por uma taxa de entrada. O que seria uma taxa de entrada? Taxa de entrada representa quantos veículos entram no sistema em um certo período de tempo. Agora imagine uma taxa de entrada de 4 veículos/min, esta taxa representa que 4 veículos entram no sistema em 1 minuto. Após 10 minutos cerca de 40 veículos entraram no sistema. Neste modelo os veículos são

chamados de população de clientes (Population of Customers), esta população tem um objetivo final, passar pelo ponto de intersecção onde há um controlador de trânsito e seguir o seu caminho na via. O servidor (Server) é representado pelo semáforo e é ele que vai servir seus clientes no modelo. Para a saída de veículos há uma taxa semelhante a taxa de entrada, porém agora de saída. O que seria a taxa de saída? A taxa de saída é um valor que diz quantos veículos sairão do sistema por um certo período de tempo. Se o sistema apresenta uma taxa de saída de 2 veículos/min, isto quer dizer que em 1 minuto 2 veículos saíram do sistema. Agora analisando uma situação com estes valores de taxa de entrada e taxa de saída apresentados anteriormente. Neste sistema está com a fila vazia e houve uma taxa de entrada de 4 veículos/min e uma taxa de saída de 2 veículos/min. Analisando o período de 1 minuto, entrou 4 veículos (enqueue na fila) e saiu 2 veículos (dequeue na fila), no final de 1 minuto o size da fila será de 2 veículos, pois o server conseguiu servir 2 clientes da fila e os outros dois não conseguiram ser servidos a tempo e então terão que aguardar a próxima abertura de semáforo. Um cliente só conseguirá sair do sistema quando ele for servido pelo server(semáforo).

Este projeto apresenta 3 modelos próprios de controle de tráfego fazendo o uso de toda a explicação acima da modelagem do sistema com taxas de entrada e saída, o que difere um modelo do outro é a política adotada e todos eles serão explicados a seguir.

Modelo 1 – Tamanho de Fila: O modelo 1 apresenta uma política de abertura e fechamento de semáforo baseado no maior tamanho de fila. O que isto quer dizer? Isto representa uma política, a qual um semáforo real teria a possibilidade de contar quantos veículos automotores há no sistema(fila) e comparar o tamanho de fila da outra rua em questão. Ele faria uma comparação se size.FilaA()>size.FilaB(), então o controlador deixaria verde para a fila A até que a fila B for maior que a fila A.

Modelo 2 - Aleatório: O modelo 2 apresenta uma política de abertura e fechamento de semáforo que não é encontrada em um semáforo de rua por exemplo. Mas foi decidido a utilização da mesma com o intuito de comparação entre os modelos em questão. A política é baseada em um sorteio realizado pelo controlador de trânsito. S e for sorteado 1 o controlador deixará aberto para a rua A, se for sorteado 2 o semáforo ficará aberto para a rua B. Basicamente a abertura e fechamento será decido aleatoriamente.

Modelo 3 – Tempo fixo de aberto: O modelo 3 apresenta uma política de abertura e fechamento a qual haverá um tempo fixo igual de semáforo aberto para a rua A e B. Se for decidido que os semáforos ficarão 30 períodos de tempo aberto, quer dizer que o semáforo A ficará 30 períodos de tempo aberto e o B ficará 30 períodos de tempo fechado, e depois o B ficará 30 períodos de tempo aberto e o A 30 períodos de tempo fechado.

Já foi descrito os 3 modelos de maneira resumida e breve, apenas para ter uma noção geral quando de fato apresentarmos os algoritmos dos modelos e assim iniciarmos o tutorial de cada um deles.

Modelo 1 – Tamanho de Fila

```
//define valor inicial para fila A
#define A 0
//define valor inicial para fila B
#define B 0
//define a quantidade de períodos de simulação
#define X 240
//define o número de vezes que será simulado(tentativas)
#define Y 100

//representa a fila de veículos da rua A
int fila_A = A;
//representa a fila de veículos da rua B
int fila_B = B;
//representa a taxa de saída no sistema(veículos/período de tempo)
int taxa_saida=0;
//representa a taxa de entrada no sistema(veículos/período de tempo)
int taxa_entrada=0;
```

```
//representa a decisão tomada pelo controlador(aberto A/aberto B)
char acao[15];
//varíavel de incremento, representando cada período de tempo
int i =0;
int open_A=0,open_B=0;
//variáveis para somar a quantidade de veículos presente nas filas
// durante o período de tempo total de simulação
int soma=0,soma_b=0;
```

Foi utilizado define para valores iniciais das filas, número de períodos simulados em cada tentativa e quantas tentativas seriam realizadas neste modelo. As variáveis apresentadas nas linhas de código são variáveis globais. Nos comentários há explicação de cada uma delas. Temos as variáveis representando as filas de veículos, a taxa de entrada e saída e também a variável representando o controlador. Repare que estas variáveis estão representando aquele mesmo modelo explicado anteriormente do sistema de uma fila, clientes, servidor, 1 entrada, 1 saída.

Agora será apresentado as funções desenvolvidas para este modelo:

```
//função de proteção, evitar fila com valor negativo(A)
int fila_vazia_A(){
        if(fila_A<0)
                fila_A=0;
        return fila_A;
}
//função de proteção, evitar fila com valor negativo(B)
int fila_vazia_B(){
        if(fila_B<0)
                fila_B=0;
        return fila_B;
        }
```

Esta função evita que a fila de veículos fique com valores negativos. É uma proteção para evitar possíveis problemas de resultados indesejáveis.

```
//Gera um taxa de entrada com valores aleatórios entre 0-4(veículos/período de tempo)
void gerador_veiculo(){
```

```
	taxa_entrada = (rand()%5);
	}
```

Aqui encontramos a função que gera veículos no sistema. Basicamente ela sorteia valores entre 0 a 4. Estes valores serão a taxa de entrada do sistema. Se sortear o valor 2, representará 2 veículos/período entraram no sistema. Ou seja, após 1 minutos 2 veículos entraram no sistema.

```
/*Gera um taxa de saída com valores aleatórios entre 0-9(veículos/período de tempo)
 * Primeiramente é sorteado um nível de tempo de reação dos motoristas em questão
 * Se o nível sorteedo for 0 então a taxa de saída do sistema será entre 8-9(veículos/período de tempo)
 * Se o nível sorteado for 1 então a taxa de saída do sistema será entre 6-7(veículos/período de tempo)
 * Se o nível sorteado for 2 então a taxa de saída do sistema será entre 4-5(veículos/período de tempo)
 * Se o nível sorteado for 3 então a taxa de saída do sistema será entre 2-3(veículos/período de tempo)
 * Se o nível sorteado for 4 então a taxa de saída do sistema será entre 0-1(veículos/período de tempo)
 */
void taxa_saida_veiculo(){
	int temp_rea = rand()%5;
	if(temp_rea==0)
		taxa_saida = rand()%10+8;
	else if(temp_rea==1)
		taxa_saida = rand()%8+6;
	else if(temp_rea==2)
		taxa_saida = rand()%6+4;
```

```
        else if(temp_rea==3)
                taxa_saida = rand()%4+2;
        else
                taxa_saida = rand()%2;

        }
```

Esta função representa a taxa de saída do sistema. Ela leva em conta o tempo de reação médio dos motoristas. Primeira mente é sorteado um número entre 0 a 4. Este número representa a nível do tempo de reação dos motoristas do sistema. Se for nível zero quer dizer que os motoristas tiveram um excelente tempo de reação, e assim a taxa de saída estará numa faixa de valores maiores desta função. Ela estará entre 8 e 9. Com o aumento de nível de tempo de reação a taxa de saída tende a diminuir. Concluindo, o valor de nível de tempo de reação e inversamente proporcional a taxa de saída de veículos. Quanto maior o nível de tempo de reação menor será a taxa de saída.

```
/*Esta função observa o ambiente das duas filas
 * e verifica qual fila é maior e então returna um valor de acordo
 * com a observação
 * Caso as filas forem iguais então ele retorna 3
 * Esta informação retornada será utilizada no controlador de
 * trânsito
 */
int ambiente_observacao(){
        if(fila_A>fila_B){
                return 1;
        }

        else if(fila_B>fila_A)
                return 0;

        else
                return 3;
        }
```

Esta função representa um possível ambiente de observação. O ambiente irá observar os sistemas e identificará o sistema com o maior size de fila, e então irá retornar um valor 1 se a fila A for a maior, 0 se a fila B for a maior ou 3 se elas forem iguais. Estas informações serão de extrema importância para o controlador de trânsito.

```
/* O controlador recebe uma informação do ambiente de observação
 * A informação é referente ao tamanho das filas das ruas(A e B)
 * Se a obs == 1 então quer dizer que a fila_A>fila_B, o controlador decide deixar
 * aberto o samáforo A neste momento
 * Se a obs == 2 então quer dizer que a fila_B>fila_A, o controlador decide deixar
 * aberto o samáforo B neste momento
 * Caso obs == 3 quer dizer que as filas são iguais então o controlador
 * realiza um sorteio para decidir qual semáforo estará aberto neste momento
 */
char*  controlador(int obs){

        if(obs==3){
                obs = (rand()%2);
        }

        if(obs==1){
```

```
            return "aberto A";
        }
        else {
            return "aberto B";
        }



        }
```

Esta função representa o controlador de trânsito. O controlador recebe a informação do ambiente de observação. Caso os tamanhos de fila forem iguais ele sorteia um número entre 0 e 1. Se a observação diz que a fila A é a maior então o controlador de trânsito irá decidir: “Aberto A”, caso seja a B a sua decisão será: “Aberto B”. A função do controlador irá retornar a decisão do controlador em uma string.

```
/*Este ambiente representa a entrada de veículos no sistema(taxa de entrada) e
 * a saída de veículos do sistema(taxa de sáida), as taxas reresentam veículos/período de tempo
 * O ambiente de simulação recebe char*, o qual é decisão do controlador
 * Quando o semáforo está aberto para a fila de veículos A Há taxa de entrada e saída de veículos
do sistema
 * representado pela fila A(veículos/período de tempo) e no sistema da fila B há apenas
 * taxa de entrada de veículos(veículos/período de tempo)
 *  * Quando o semáforo está aberto para a fila de veículos B Há taxa de entrada e saída de
veículos do sistema
 * representado pela fila B(veículos/período de tempo) e no sistema da fila A há apenas
 * taxa de entrada de veículos(veículos/período de tempo)
 *
 */
int ambiente_simul(char* acao){
        if (strcmp(acao,"aberto A")==0){
```

```
/*Se a fila A está vazia
 * ela terá apenas taxa de entrada de veículos
 * e a fila B terá também taxa de entrada de
 * veículos
 */
if(fila_A<0){
        /*É gerado uma taxa de entrada
         */
        gerador_veiculo();
        /*É adicionado veículos por período na fila A
         */
        fila_A += taxa_entrada;
        /*Varíavel para calcular a média de veículos no total de períodos
        de simulação na fila A
         */
        soma = soma + fila_A;
        /*É gerado uma taxa de entrada
         */
        gerador_veiculo();
        /*É adicionado veículos por período na fila B
         */
        fila_B +=taxa_entrada;
        /*Varíavel para calcular a média de veículos no total de períodos
        de simulação na fila B
         */
        soma_b = soma_b + fila_B;
        i++;


        }
```

A função ambiente de simulação irá simular a situação descrita para aquela imagem apresentada anteriormente de um cruzamento em X. Esta função recebe a decisão retornada pela função do controlador e assim executa a simulação. Caso aquele char pointer for uma string “Aberto A” então a rua A terá o semáforo verde para a circulação de veículos na via. Dentro do IF “Aberto A” é testado se a fila primeiro se a fila estiver vazia, apenas será gerado veículo aos sistemas da fila aberta, ou seja, taxa de entrada. Por isso a função gerador_veiculo é chamada para colocar veículos no sistema naquele período de tempo. Assim de acordo com o valor sorteado pela função gerador_veiculo é assim adicionado na fila de veículos A. Como o semáforo B está fechado, não há a possibilidade de veículos saírem da fila B, então enquanto o semáforo A estiver aberto há taxa de entrada

na fila B, ou seja, veículos estão sendo adicionados na fila B. É utilizado variáveis para cálculos de futuras médias.

```
else{
                        /*É gerado uma taxa de saída
                         */
                        taxa_saida_veiculo();
                        /*É retirado veículos por período na fila A
                         */
                        fila_A -= taxa_saida;
                        fila_vazia_A();
                        /*Varíavel para calcular a média de veículos no total de períodos
                        de simulação na fila A
                         */

                        gerador_veiculo();
                        /*É adicionado veículos por período na fila A
                         */
                        fila_A += taxa_entrada;
                        /*Varíavel para calcular a média de veículos no total de períodos
                                                de simulação na fila A
                         */
                        soma = soma + fila_A;
                        /*É gerado uma taxa de entrada
                         */
                        gerador_veiculo();
                        /*É adicionado veículos por período na fila B
                         */
                        fila_B +=taxa_entrada;
                        /*Varíavel para calcular a média de veículos no total de períodos
                        de simulação na fila B
                         */
                        soma_b = soma_b + fila_B;
                        i++;

                }
                open_A++;
```

Este Else representa o Aberto A ainda, mas agora com condição de final não vazia. Em resumo o que será feito aqui em ordem: Retirar veículos da fila A, é gerado uma taxa de saída para este período, de acordo com este valor é retirado da fila. Após a retirada é feita a proteção de fila vazia, para evitar valores negativos para a fila de veículos. Para fila B é gerado uma taxa de entrada pela função gerado_veiculo e assim é adicionado veículos na fila B. Quando um semáforo está aberto há taxa de entrada e saída no semáforo aberto e apenas

taxa de entrada no semáforo fechado. E somado os valores das filas naquele período para um cálculo de uma média futura.

Agora é quando o semáforo se encontra aberto para a fila B. Se a string presente na acao for igual a “aberto B”, quer dizer que o controlador decidiu deixar aberto para o semáforo B.

```
else if(strcmp(acao,"aberto B")==0){
                if(fila_B<0){
                        /*É gerado uma taxa de entrada
                         */
                        gerador_veiculo();
                        /*É adicionado veículos por período na fila A
                         */
                        fila_B += taxa_entrada;
                        /*Varíavel para calcular a média de veículos no total de períodos
                                                de simulação na fila A
                         */
                        soma_b = soma_b + fila_B;
                        /*É gerado uma taxa de entrada
                         */
                        gerador_veiculo();
                        /*É adicionado veículos por período na fila B
                         */
                        fila_A +=taxa_entrada;
                        /*Varíavel para calcular a média de veículos no total de períodos
                        de simulação na fila B
                         */
                        soma = soma + fila_A;
                        i++;


                }
                else{
                        /*É gerado uma taxa de saída
                         */
                        taxa_saida_veiculo();
                        /*É retirado veículos por período na fila A
                         */
                        fila_B -= taxa_saida;
                        fila_vazia_B();
                        /*Varíavel para calcular a média de veículos no total de períodos
                        de simulação na fila A
                         */



                        gerador_veiculo();
                        /*É adicionado veículos por período na fila A
```

```
                    */
                    fila_B += taxa_entrada;
                    /*Varíavel para calcular a média de veículos no total de períodos
                                                                de simulação na fila A
                    */
                    soma_b = soma_b + fila_B;
                    /*É gerado uma taxa de entrada
                    */
                    gerador_veiculo();
                    /*É adicionado veículos por período na fila B
                    */
                    fila_A +=taxa_entrada;
                    /*Varíavel para calcular a média de veículos no total de períodos
                                        de simulação na fila B
                    */
                    soma = soma + fila_A;
                    i++;
            }
            open_B++;
    }
```

O funcionamento deste código é idêntico ao código apresentada para quando o semáforo A estava aberto, o que muda é que o semáforo B está aberto neste momento, então quando um semáforo está aberto há taxa de entrada e saída para o semáforo aberto e apenas taxa de sáida para o semáforo fechado.

```
int main(void) {
        int sa=0,sb=0;
        int  obs = 0,h=0;
        //semente para gerar números aleatórios
        srand(time(NULL));
```

Estas foram as variáveis utilizadas na função main. Srand(time(NULL)) representa a semente para o uso da função rand(). Para evitar que cada vez que o programa é simulado e sorteie valores distintos.

```
        while(h!=Y){
                soma = 0;
                soma_b = 0;
                i =0;
                gerador_veiculo();
                fila_A += taxa_entrada;
                soma = soma + fila_A;
                gerador_veiculo();
                soma_b = soma_b + fila_B;
```

```
//printf("Fila_A=%d,Fila_B=%d\n",fila_A,fila_B);
```

Este primeiro while é o laço de fora, cada iteração dele será uma tentativa de simulação para um número de períodos estipulado. É gerado veículos para os sistemas, para as suas filas.

```
while(i!=X){
        //verificar se as filas estão vazias
        fila_vazia_A();
        fila_vazia_B();
        //observar o ambiente
        obs = ambiente_observacao();
        /*eviar a observação ao controlador e o
         * controlador irá decidir qual semáforo deixará aberto
         * ambiente_simul simula a situação do cruzamento em X
         */
        ambiente_simul(controlador(obs));
        //verifica se as filas estão vazias
        fila_vazia_A();
        fila_vazia_B();
        //printf("Tempo = %d, situação %s, A = %d,B =
%d\n",i,controlador(obs),fila_A,fila_B);
```

Este while representa a simulação neste caso durante 240 períodos de tempo, podendo ser alterado. Este é o laço de dentro. Basicamente nele é verificado se ambas as filas apresentam um número negativo para seu valor de fila, caso apresentam é atribuído zero para seu valor. Após esta etapa é preciso fazer uma observação no ambiente dos sistemas e assim o ambiente retornará um valor para o controlador de trânsito. O controlador decidirá deixar aberto para o sistema com maior tamanho de fila. O ambiente de simulação receberá a string retornada pelo controlador e é lá que o cruzamento em X é simulado. O sistema como um todo. Esse laço durará o número de período estipulado pelo usuário. Neste caso 240 períodos.

```
}
//calcular a média de vaículos da fila A em relação ao total de perídos simulados
```

```
            sa+= soma/i;
            //calcular a média de vaículos da fila A em relação ao total de perídos simulados
            sb +=soma_b/i;
            fila_A = A;
            fila_B = B;
```

Neste caso está sendo usado uma variável para o cálculo de uma média. E as filas serão esvaziadas para repetir novamente a simulação para averiguar o próximo resultado.

```
            h++;
            //impressão da média de veículos das filas em cada tentativa de simulação
            printf("Tentativa( %d)  Média Veículo A: %f,Média Veiculo B: %f por %d periodos de tempo\n",h,(float)soma/i,(float)soma_b/i,X);
```

É apresentado a média de veículos para cada fila nas tentativas de simulação. No caso foi feito 100 tentativas para simulações de 240 períodos de tempo.

```
        }
        //impressão da média de veículos nas filas em relação a todas as tentativas
        printf("Média Veículo A: %f,Média Veiculo B: %f por %d períodos de tempo em %d tentativas\n",(float)sa/h,(float)sb/h,X,Y);
        return 0;

}
```

No final foi feito uma média entre todas as tentativas de simulação.

Modelo 2 – Aleatório

```
//define valor inicial para fila A
#define A 0
//define valor inicial para fila B
#define B 0
//define a quantidade de períodos de simulação
#define X 240
//define o número de vezes que será simulado(tentativas)
#define Y 100

//representa a fila de veículos da rua A
int fila_A = A;
```

```
//representa a fila de veículos da rua B
int fila_B = B;
//representa a taxa de saída no sistema(veículos/período de tempo)
int taxa_saida=0;
//representa a taxa de entrada no sistema(veículos/período de tempo)
int taxa_entrada=0;
//representa a decisão tomada pelo controlador(aberto A/aberto B)
char acao[15];
//varíavel de incremento, representando cada período de tempo
int i =0;
int open_A=0,open_B=0;
//variáveis para somar a quantidade de veículos presente nas filas
// durante o período de tempo totald e simulação
int soma=0,soma_b=0;
```

Foi utilizado define para valores iniciais das filas, número de períodos simulados em cada tentativa e quantas tentativas seriam realizadas neste modelo. As variáveis apresentadas nas linhas de código são variáveis globais. Nos comentários há explicação de cada uma delas. Temos as variáveis representando as filas de veículos, a taxa de entrada e saída e também a variável representando o controlador. Repare que estas variáveis estão representando aquele mesmo modelo explicado anteriormente do sistema de uma fila, clientes, servidor, 1 entrada, 1 saída.

Agora será apresentado as funções desenvolvidas para este modelo:

```
//função de proteção, evitar fila com valor negativo(A)
int fila_vazia_A(){
        if(fila_A<0)
                fila_A=0;
        return fila_A;
}
//função de proteção, evitar fila com valor negativo(B)
int fila_vazia_B(){
        if(fila_B<0)
                fila_B=0;
        return fila_B;
```

```
}
```

Esta função evita que a fila de veículos fique com valores negativos. É uma proteção para evitar possíveis problemas de resultados indesejáveis.

```
//Gera um taxa de entrada com valores aleatórios entre 0-4(veículos/período de tempo)
void gerador_veiculo(){

	taxa_entrada = (rand()%5);
}
```

Aqui encontramos a função que gera veículos no sistema. Basicamente ela sorteia valores entre 0 a 4. Estes valores serão a taxa de entrada do sistema. Se sortear o valor 2, representará 2 veículos/período entraram no sistema. Ou seja, após 1 minutos 2 veículos entraram no sistema.

```
/*Gera um taxa de saída com valores aleatórios entre 0-9(veículos/período de tempo)
 * Primeiramente é sorteado um nível de tempo de reação dos motoristas em questão
 * Se o nível sorteedo for 0 então a taxa de saída do sistema será entre 8-9(veículos/período de tempo)
 * Se o nível sorteado for 1 então a taxa de saída do sistema será entre 6-7(veículos/período de tempo)
 * Se o nível sorteado for 2 então a taxa de saída do sistema será entre 4-5(veículos/período de tempo)
 * Se o nível sorteado for 3 então a taxa de saída do sistema será entre 2-3(veículos/período de tempo)
 * Se o nível sorteado for 4 então a taxa de saída do sistema será entre 0-1(veículos/período de tempo)
 */
void taxa_saida_veiculo(){
	int temp_rea = rand()%5;
	if(temp_rea==0)
		taxa_saida = rand()%10+8;
	else if(temp_rea==1)
		taxa_saida = rand()%8+6;
	else if(temp_rea==2)
		taxa_saida = rand()%6+4;
	else if(temp_rea==3)
		taxa_saida = rand()%4+2;
	else
		taxa_saida = rand()%2;

}
```

Esta função representa a taxa de saída do sistema. Ela leva em conta o tempo de reação médio dos motoristas. Primeira mente é sorteado um número entre 0 a 4. Este número representa a nível do tempo de reação dos motoristas do

sistema. Se for nível zero quer dizer que os motoristas tiveram um excelente tempo de reação, e assim a taxa de saída estará numa faixa de valores maiores desta função. Ela estará entre 8 e 9. Com o aumento de nível de tempo de reação a taxa de saída tende a diminuir. Concluindo, o valor de nível de tempo de reação e inversamente proporcional a taxa de saída de veículos. Quanto maior o nível de tempo de reação menor será a taxa de saída.

```
/*É realizado um sorteio para a abertura
 * dos semáforos das ruas A e B
 */
int ambiente_observacao(){
	int obs = (rand()%2);
	return obs;
}
```

Ambiente de observação irá sortear um número aleatório entre 0 e 1 para enviar para o controlador. Já que este modelo tem a política de abertura e fechamento do semáforo de maneira aleatória. Este ambiente representa esta política.

```
/*De acordo com o sorteio o controlador
 * irá abrir o samáforo A ou B
 */
char*  controlador(int obs){

	if(obs==1){
		return "aberto A";
	}
	else
		return "aberto B";


}
```

De acordo com o sorteio do ambiente de observação o controlador vai deixa aberto para o semáforo A caso a obs ser 1 ou aberto para o semáforo B caso a obs for 0.

```
/*Este ambiente representa a entrada de veículos no sistema(taxa de entrada) e
 * a saída de veículos do sistema(taxa de sáida), as taxas reresentam veículos/período de tempo
 * O ambiente de simulação recebe char*, o qual é decisão do controlador
```

```
 * Quando o semáforo está aberto para a fila de veículos A Há taxa de entrada e saída de veículos
do sistema
 * representado pela fila A(veículos/período de tempo) e no sistema da fila B há apenas
 * taxa de entrada de veículos(veículos/período de tempo)
 *  * Quando o semáforo está aberto para a fila de veículos B Há taxa de entrada e saída de
veículos do sistema
 * representado pela fila B(veículos/período de tempo) e no sistema da fila A há apenas
 * taxa de entrada de veículos(veículos/período de tempo)
 *
 */
int ambiente_simul(char* acao){
        if (strcmp(acao,"aberto A")==0){
                /*Se a fila A está vazia
                 * ela terá apenas taxa de entrada de veículos
                 * e a fila B terá também taxa de entrada de
                 * veículos
                 */
                if(fila_A<0){
                        /*É gerado uma taxa de entrada
                         */
                        gerador_veiculo();
                        /*É adicionado veículos por período na fila A
                         */
                        fila_A += taxa_entrada;
                        /*Varíavel para calcular a média de veículos no total de períodos
                        de simulação na fila A
                         */
                        soma = soma + fila_A;
                        /*É gerado uma taxa de entrada
                         */
                        gerador_veiculo();
                        /*É adicionado veículos por período na fila B
                         */
                        fila_B +=taxa_entrada;
                        /*Varíavel para calcular a média de veículos no total de períodos
                        de simulação na fila B
                         */
                        soma_b = soma_b + fila_B;
                        i++;


                }
```

A função ambiente de simulação irá simular a situação descrita para aquela imagem apresentada anteriormente de um cruzamento em X. Esta função recebe a decisão retornada pela função do controlador e assim executa a simulação. Caso aquele char pointer for uma string “Aberto A” então a rua A terá o semáforo verde para a circulação de veículos na via. Dentro do IF “Aberto A” é testado se a fila primeiro se a fila estiver vazia, apenas será gerado veículo aos sistemas da fila

aberta, ou seja, taxa de entrada. Por isso a função gerador_veiculo é chamada para colocar veículos no sistema naquele período de tempo. Assim de acordo com o valor sorteado pela função gerador_veiculo é assim adicionado na fila de veículos A. Como o semáforo B está fechado, não há a possibilidade de veículos saírem da fila B, então enquanto o semáforo A estiver aberto há taxa de entrada na fila B, ou seja, veículos estão sendo adicionados na fila B. É utilizado variáveis para cálculos de futuras médias.

```
else{
                        /*É gerado uma taxa de saída
                         */
                        taxa_saida_veiculo();
                        /*É retirado veículos por período na fila A
                         */
                        fila_A -= taxa_saida;
                        fila_vazia_A();
                        /*Varíavel para calcular a média de veículos no total de períodos
                        de simulação na fila A
                         */

                        gerador_veiculo();
                        /*É adicionado veículos por período na fila A
                         */
                        fila_A += taxa_entrada;
                        /*Varíavel para calcular a média de veículos no total de períodos
                                                de simulação na fila A
                         */
                        soma = soma + fila_A;
                        /*É gerado uma taxa de entrada
                         */
                        gerador_veiculo();
                        /*É adicionado veículos por período na fila B
                         */
                        fila_B +=taxa_entrada;
                        /*Varíavel para calcular a média de veículos no total de períodos
                        de simulação na fila B
                         */
                        soma_b = soma_b + fila_B;
                        i++;

                }
                open_A++;

        }
```

Este Else representa o Aberto A ainda, mas agora com condição de final não vazia. Em resumo o que será feito aqui em ordem: Retirar veículos da fila A, é

gerado uma taxa de saída para este período, de acordo com este valor é retirado da fila. Após a retirada é feita a proteção de fila vazia, para evitar valores negativos para a fila de veículos. Para fila B é gerado uma taxa de entrada pela função gerador_veiculo e assim é adicionado veículos na fila B. Quando um semáforo está aberto há taxa de entrada e saída no semáforo aberto e apenas taxa de entrada no semáforo fechado. E somado os valores das filas naquele período para um cálculo de uma média futura.

Agora é quando o semáforo se encontra aberto para a fila B. Se a string presente a acao for igual a “aberto B”, quer dizer que o controlador decidiu deixar aberto para o semáforo B.

```
else if(strcmp(acao,"aberto B")==0){
                    if(fila_B<0){
                              /*É gerado uma taxa de entrada
                               */
                              gerador_veiculo();
                              /*É adicionado veículos por período na fila A
                               */
                              fila_B += taxa_entrada;
                              /*Varíavel para calcular a média de veículos no total de períodos
                                                  de simulação na fila A
                               */
                              soma_b = soma_b + fila_B;
                              /*É gerado uma taxa de entrada
                               */
                              gerador_veiculo();
                              /*É adicionado veículos por período na fila B
                               */
                              fila_A +=taxa_entrada;
                              /*Varíavel para calcular a média de veículos no total de períodos
                              de simulação na fila B
                               */
                              soma = soma + fila_A;
                              i++;


                    }
                    else{
                              /*É gerado uma taxa de saída
                               */
                              taxa_saida_veiculo();
                              /*É retirado veículos por período na fila A
                               */
                              fila_B -= taxa_saida;
```

```
                    fila_vazia_B();
                    /*Varíavel para calcular a média de veículos no total de períodos
                    de simulação na fila A
                     */

                    gerador_veiculo();
                    /*É adicionado veículos por período na fila A
                     */
                    fila_B += taxa_entrada;
                    /*Varíavel para calcular a média de veículos no total de períodos
                                                                de simulação na fila A
                     */
                    soma_b = soma_b + fila_B;
                    /*É gerado uma taxa de entrada
                     */
                    gerador_veiculo();
                    /*É adicionado veículos por período na fila B
                     */
                    fila_A +=taxa_entrada;
                    /*Varíavel para calcular a média de veículos no total de períodos
                                          de simulação na fila B
                     */
                    soma = soma + fila_A;
                    i++;
            }
            open_B++;
    }
```

O funcionamento deste código é idêntico ao código apresentada para quando o semáforo A estava aberto, o que muda é que o semáforo B está aberto neste momento, então quando um semáforo está aberto há taxa de entrada e saída para o semáforo aberto e apenas taxa de sáida para o semáforo fechado.

```
int main(void) {
        int sa=0,sb=0;
        int  obs = 0,h=0;

//semente para gerar números aleatórios
        srand(time(NULL));
```

Estas foram as variáveis utilizadas na função main. Srand(time(NULL)) representa a semente para o uso da função rand(). Para evitar que cada vez que o programa é simulado e sorteie valores distintos.

```
while(h!=Y){
        soma = 0;
        soma_b = 0;

        i =0;
        gerador_veiculo();
        fila_A += taxa_entrada;
        soma = soma + fila_A;
        gerador_veiculo();
        soma_b = soma_b + fila_B;



        //printf("Fila_A=%d,Fila_B=%d\n",fila_A,fila_B);
```

Este primeiro while é o laço de for, cada iteração dele será uma tentativa de simulação para um número de períodos estipulado. É gerado veículos para os sistemas, para as suas filas.

```
        while(i!=X){

                //verificar se as filas estão vazias
                fila_vazia_A();
                fila_vazia_B();
                //observar o ambiente
                obs = ambiente_observacao();
                /*eviar a observação ao controlador e o
                 * controlador irá decidir qual semáforo deixará aberto
                 * ambiente_simul simula a situação do cruzamento em X
                 */
                ambiente_simul(controlador(obs));
                //verifica se as filas estão vazias
                fila_vazia_A();
                fila_vazia_B();
                //printf("Tempo = %d, situação %s, A = %d,B =
%d\n",i,controlador(obs),fila_A,fila_B);
```

Este while representa a simulação neste caso durante 240 períodos de tempo, podendo ser alterado. Este é o laço de dentro. Basicamente nele é verificado se ambas as filas apresentam um número negativo para seu valor de fila, caso apresentam é atribuído zero para seu valor. Após esta etapa é preciso fazer uma observação no ambiente dos sistemas e assim o ambiente retornará um

valor para o controlador de trânsito. O controlador decidirá deixar aberto para o sistema com maior tamanho de fila. O ambiente de simulação receberá a string retornada pelo controlador e é lá que o cruzamento em X é simulado. O sistema como um todo. Esse laço durará o número de período estipulado pelo usuário. Neste caso 240 períodos.

```
}
//calcular a média de vaículos da fila A em relação ao total de perídos simulados
sa+= soma/i;
//calcular a média de vaículos da fila A em relação ao total de perídos simulados
sb +=soma_b/i;
fila_A = A;
fila_B = B;
```

Neste caso está sendo usado uma variável para o cálculo de uma média. E as filas serão esvaziadas para repetir novamente a simulação para averiguar o próximo resultado.

```
h++;
//impressão da média de veículos das filas em cada tentativa de simulação
printf("Tentativa( %d)  Média Veículo A: %f,Média Veiculo B: %f por %d periodos de tempo\n",h,(float)soma/i,(float)soma_b/i,X);
```

É apresentado a média de veículos para cada fila nas tentativas de simulação. No caso foi feito 100 tentativas para simulações de 240 períodos de tempo.

```
}
//impressão da média de veículos nas filas em relação a todas as tentativas
printf("Média Veículo A: %f,Média Veiculo B: %f por %d períodos de tempo em %d tentativas\n",(float)sa/h,(float)sb/h,X,Y);
return 0;

}
```

No final foi feito uma média entre todas as tentativas de simulação.

Modelo 3 – Tempo Fixo de Aberto

```
//define valor inicial para fila A
#define A 0
//define valor inicial para fila B
#define B 0
//define a quantidade de períodos de simulação
#define X 240
//define o número de vezes que será simulado(tentativas)
#define Y 100
//define período de semaforo A aberto
#define K 25
//define período de semáforo B aberto
#define Z 15


//representa a fila de veículos da rua A
int fila_A = A;
//representa a fila de veículos da rua B
int fila_B = B;
//representa a taxa de saída no sistema(veículos/período de tempo)
int taxa_saida=0;
//representa a taxa de entrada no sistema(veículos/período de tempo)
int taxa_entrada=0;
//representa a decisão tomada pelo controlador(aberto A/aberto B)
char acao[15];
int open_A=0,open_B=0;
```

Foi utilizado define para valores iniciais das filas, número de períodos simulados em cada tentativa e quantas tentativas seriam realizadas neste modelo. As variáveis apresentadas nas linhas de código são variáveis globais. Nos comentários há explicação de cada uma delas. Temos as variáveis representando as filas de veículos, a taxa de entrada e saída e também a variável representando o controlador. Repare que estas variáveis estão representando aquele mesmo modelo explicado anteriormente do sistema de uma fila, clientes, servidor, 1 entrada, 1 saída.

Agora será apresentado as funções desenvolvidas para este modelo:

```
//função de proteção, evitar fila com valor negativo(A)
int fila_vazia_A(){
        if(fila_A<0)
                fila_A=0;
        return fila_A;
}
//função de proteção, evitar fila com valor negativo(B)
int fila_vazia_B(){
        if(fila_B<0)
                fila_B=0;
        return fila_B;
        }
```

Esta função evita que a fila de veículos fique com valores negativos. É uma proteção para evitar possíveis problemas de resultados indesejáveis.

```
//Gera um taxa de entrada com valores aleatórios entre 0-4(veículos/período de tempo)
void gerador_veiculo(){

        taxa_entrada = (rand()%5);
}
```

Aqui encontramos a função que gera veículos no sistema. Basicamente ela sorteia valores entre 0 a 4. Estes valores serão a taxa de entrada do sistema. Se sortear o valor 2, representará 2 veículos/período entraram no sistema. Ou seja, após 1 minutos 2 veículos entraram no sistema.

```
/*Gera um taxa de saída com valores aleatórios entre 0-9(veículos/período de tempo)
 * Primeiramente é sorteado um nível de tempo de reação dos motoristas em questão
 * Se o nível sorteedo for 0 então a taxa de saída do sistema será entre 8-9(veículos/período de tempo)
 * Se o nível sorteado for 1 então a taxa de saída do sistema será entre 6-7(veículos/período de tempo)
 * Se o nível sorteado for 2 então a taxa de saída do sistema será entre 4-5(veículos/período de tempo)
 * Se o nível sorteado for 3 então a taxa de saída do sistema será entre 2-3(veículos/período de tempo)
 * Se o nível sorteado for 4 então a taxa de saída do sistema será entre 0-1(veículos/período de tempo)
 */
void taxa_saida_veiculo(){
```

```
        int temp_rea = rand()%5;
        if(temp_rea==0)
                taxa_saida = rand()%10+8;
        else if(temp_rea==1)
                taxa_saida = rand()%8+6;
        else if(temp_rea==2)
                taxa_saida = rand()%6+4;
        else if(temp_rea==3)
                taxa_saida = rand()%4+2;
        else
                taxa_saida = rand()%2;

}
```

Esta função representa a taxa de saída do sistema. Ela leva em conta o tempo de reação médio dos motoristas. Primeira mente é sorteado um número entre 0 a 4. Este número representa a nível do tempo de reação dos motoristas do sistema. Se for nível zero quer dizer que os motoristas tiveram um excelente tempo de reação, e assim a taxa de saída estará numa faixa de valores maiores desta função. Ela estará entre 8 e 9. Com o aumento de nível de tempo de reação a taxa de saída tende a diminuir. Concluindo, o valor de nível de tempo de reação e inversamente proporcional a taxa de saída de veículos. Quanto maior o nível de tempo de reação menor será a taxa de saída.

```
int main(void) {
        int m=0;
        //variáveis para somar a quantidade de veículos presente nas filas
        // durante o período de tempo total de simulação
        int soma=0,soma_b=0;
        //variáveis para somar a as médias de veículos presente nas filas
        //durante todas tentativa de simulação
        int sa=0,sb=0;
        //variável de incrente do total de tentativas de simulação
        int h=0;
        //semente para gerar números aleatórios
        srand(time(NULL));
```

```
while(h!=Y){

        gerador_veiculo();
        fila_A += taxa_entrada;
        soma = soma + fila_A;
        gerador_veiculo();
        fila_B += taxa_entrada;
        soma_b = soma_b + fila_B;
        m = 0;

        //printf("Fila_A=%d,Fila_B=%d\n",fila_A,fila_B);
        while(m!=X){



                int j = K;
                while(j!=0){
                        /*É gerado uma taxa de saída
                         */
                        taxa_saida_veiculo();
                        /*É retirado veículos por período na fila A
                         */
                        fila_A -= taxa_saida;
                        /*Varíavel para calcular a média de veículos no total de períodos
                        de simulação na fila A
                         */
                        fila_vazia_A();



                        gerador_veiculo();
                        /*É adicionado veículos por período na fila A
                         */
                        fila_A += taxa_entrada;
                        /*Varíavel para calcular a média de veículos no total de períodos
                        de simulação na fila A
                         */
                        soma = soma + fila_A;
                        /*É gerado uma taxa de entrada
                         */
                        gerador_veiculo();
                        /*É adicionado veículos por período na fila B
                         */
                        fila_B +=taxa_entrada;
                        /*Varíavel para calcular a média de veículos no total de períodos
                        de simulação na fila B
```

```
                */
                soma_b = soma_b + fila_B;
                j--;
                m++;
                if(m>=X){
                        m = X;
                        j = 0;
                }
```

Este while representa o Aberto A. O semáforo A ficará aberto por um período fixo. Em resumo o que será feito aqui em ordem: Retirar veículos da fila A, é gerado uma taxa de saída para este período, de acordo com este valor é retirado da fila. Após a retirada é feita a proteção de fila vazia, para evitar valores negativos para a fila de veículos. Para fila B é gerado uma taxa de entrada pela função gerador_veiculo e assim é adicionado veículos na fila B. Quando um semáforo está aberto há taxa de entrada e saída no semáforo aberto e apenas taxa de entrada no semáforo fechado. E somado os valores das filas naquele período para um cálculo de uma média futura.

```
                //printf("Tempo = %d, situação Aberto A, A = %d,B =
%d\n",m,fila_A,fila_B);
        }

        j = Z;

        while(j!=0){
                /*É gerado uma taxa de saída
                 */
                taxa_saida_veiculo();
                /*É retirado veículos por período na fila A
                 */
                fila_B -= taxa_saida;
                /*Varíavel para calcular a média de veículos no total de períodos
                de simulação na fila A
                 */
                fila_vazia_B();



                gerador_veiculo();
                /*É adicionado veículos por período na fila A
                 */
                fila_B += taxa_entrada;
                /*Varíavel para calcular a média de veículos no total de períodos
```

```
de simulação na fila A
 */
soma_b = soma_b + fila_B;
/*É gerado uma taxa de entrada
 */
gerador_veiculo();
/*É adicionado veículos por período na fila B
 */
fila_A +=taxa_entrada;
/*Varíavel para calcular a média de veículos no total de períodos
de simulação na fila B
 */
soma = soma + fila_A;
j--;
m++;
if(m>=X){
        m = X;
        j = 0;
}
```

Este while representa o Aberto B. O semáforo B ficará aberto por um período fixo. Em resumo o que será feito aqui em ordem: Retirar veículos da fila B, é gerado uma taxa de saída para este período, de acordo com este valor é retirado da fila. Após a retirada é feita a proteção de fila vazia, para evitar valores negativos para a fila de veículos. Para fila A é gerado uma taxa de entrada pela função gerador_veiculo e assim é adicionado veículos na fila A. Quando um semáforo está aberto há taxa de entrada e saída no semáforo aberto e apenas taxa de entrada no semáforo fechado. E somado os valores das filas naquele período para um cálculo de uma média futura.

```
//printf("Tempo = %d, situação Aberto B, A = %d,B = %d\n",m,fila_A,fila_B);
        }




}

sa+= soma/m;
sb +=soma_b/m;
fila_A = A;
fila_B = B;
```

Neste caso está sendo usado uma variável para o cálculo de uma média. E as filas serão esvaziadas para repetir novamente a simulação para averiguar o próximo resultado.

```
                h++;
                //impressão da média de veículos das filas em cada tentativa de simulação
                printf("Tentativa( %d)  Média Veículo A: %f,Média Veiculo B: %f por %d periodos de tempo\n",h,(float)soma/i,(float)soma_b/i,X);
```

É apresentado a média de veículos para cada fila nas tentativas de simulação. No caso foi feito 100 tentativas para simulações de 240 períodos de tempo.

```
        }
        //impressão da média de veículos nas filas em relação a todas as tentativas
        printf("Média Veículo A: %f,Média Veiculo B: %f por %d períodos de tempo em %d tentativas\n",(float)sa/h,(float)sb/h,X,Y);
        return 0;

}
```

No final foi feito uma média entre todas as tentativas de simulação.

}

A primeira coluna do csv representa o número da tentativa, segunda coluna média de veículos da fila A e terceira coluna média de veículos da fila B. Cada tentativa foi simulada em 240 períodos de tempo. Foi feito 100 tentativas

Resultados do Modelo 1 – Tamanho de Fila:

1,4.141667,4.425000
2,5.237500,5.570833
3,3.950000,4.020833
4,4.391667,4.245833
5,4.083333,3.958333
6,4.041667,4.362500
7,4.400000,4.387500
8,4.637500,4.904167
9,4.341667,3.962500
10,4.662500,5.066667
11,4.075000,4.129167
12,4.050000,3.966667
13,4.587500,4.325000
14,4.879167,4.958333
15,4.383333,4.612500
16,5.395833,5.025000
17,5.016667,4.512500
18,4.262500,4.145833
19,4.312500,4.475000
20,4.500000,4.316667
21,4.529167,4.550000
22,3.754167,3.737500
23,4.491667,4.745833
24,4.104167,4.095833
25,4.358333,4.333333
26,4.225000,4.312500
27,5.308333,5.225000
28,4.191667,4.233333
29,4.779167,5.112500
30,4.000000,3.787500
31,3.937500,4.195833
32,3.883333,4.312500
33,4.704167,4.400000
34,3.775000,4.037500
35,4.075000,4.141667
36,4.087500,4.079167
37,4.633333,4.766667
38,4.933333,4.854167
39,4.025000,4.112500
40,3.412500,3.558333
41,4.395833,4.629167
42,4.145833,4.483333
43,4.300000,4.887500
44,4.220833,4.441667
45,4.262500,3.970833
46,3.908333,3.704167
47,4.683333,5.070833
48,4.583333,4.208333
49,4.379167,4.395833
50,4.637500,4.533333

51,4.850000,4.800000
52,4.258333,4.316667
53,4.425000,4.391667
54,3.891667,3.575000
55,4.637500,4.587500
56,4.587500,4.337500
57,4.204167,4.283333
58,3.862500,3.908333
59,5.141667,4.816667
60,4.762500,5.225000
61,4.762500,4.933333
62,4.200000,4.129167
63,5.337500,5.279167
64,4.487500,4.029167
65,4.000000,3.812500
66,3.783333,3.775000
67,4.129167,4.054167
68,4.316667,4.295833
69,4.633333,4.783333
70,3.658333,3.691667
71,4.450000,4.529167
72,4.308333,4.133333
73,4.204167,4.308333
74,4.770833,4.304167
75,4.537500,4.320833
76,3.891667,3.887500
77,4.170833,3.870833
78,4.683333,4.941667
79,3.845833,3.904167
80,6.670833,6.287500
81,5.275000,5.095833
82,4.620833,4.600000
83,4.912500,4.525000
84,4.954167,4.741667
85,3.870833,4.045833
86,4.233333,4.120833
87,4.954167,5.070833
88,4.866667,4.500000
89,4.533333,4.691667
90,4.829167,5.200000
91,4.229167,3.900000
92,4.837500,4.587500
93,4.562500,4.350000
94,4.333333,4.754167
95,5.029167,5.012500
96,3.850000,4.141667
97,4.529167,4.566667
98,4.820833,4.458333
99,4.195833,4.104167
100,4.191667,4.008333

Resultados do Modelo 2 - Aleatório:

1,5.958333,8.516666
2,6.575000,10.816667
3,6.904167,10.308333
4,8.500000,6.058333
5,5.191667,6.579167
6,5.558333,5.008333
7,6.200000,5.966667
8,9.420834,9.504167
9,8.091666,9.141666
10,5.533333,10.304167
11,8.595834,10.470834
12,5.504167,7.416667
13,5.837500,13.387500
14,7.866667,7.508333
15,5.366667,8.029166
16,9.254167,6.345833
17,7.125000,9.770833
18,6.566667,6.987500
19,6.800000,8.020833
20,8.037500,4.937500
21,6.845833,7.687500
22,10.983334,9.170834
23,7.975000,5.879167
24,6.854167,7.616667
25,6.150000,11.100000
26,7.562500,6.966667
27,9.337500,6.362500
28,6.725000,7.045833
29,5.020833,7.091667
30,4.908333,6.695833
31,6.716667,4.562500
32,11.358334,4.854167
33,7.245833,7.108333
34,7.000000,6.587500
35,9.025000,6.225000
36,5.904167,16.354166
37,7.558333,9.050000
38,7.833333,8.520833
39,12.412500,5.366667
40,8.187500,6.662500
41,7.445833,6.516667
42,6.658333,7.679167
43,5.483333,7.850000
44,6.012500,12.670834
45,6.691667,7.958333
46,9.895833,7.212500
47,9.312500,6.854167
48,8.062500,6.262500
49,7.833333,11.000000
50,13.604167,6.758333

51,6.825000,7.095833
52,8.316667,8.062500
53,8.195833,10.062500
54,9.000000,5.091667
55,7.729167,8.070833
56,5.587500,5.908333
57,5.054167,10.537500
58,8.308333,10.229167
59,7.770833,7.066667
60,6.408333,6.966667
61,8.737500,5.220833
62,10.412500,6.479167
63,17.412500,8.262500
64,6.908333,8.366667
65,5.833333,8.395833
66,8.100000,7.762500
67,14.225000,7.525000
68,10.370833,7.050000
69,6.595833,8.479167
70,5.325000,8.779166
71,6.666667,6.604167
72,6.029167,9.458333
73,8.620833,8.562500
74,6.750000,5.787500
75,5.187500,8.354167
76,8.083333,11.066667
77,12.733334,4.695833
78,5.337500,9.533334
79,6.029167,11.341666
80,5.762500,7.304167
81,6.666667,6.720833
82,10.645833,5.779167
83,8.395833,6.245833
84,7.104167,6.195833
85,8.462500,6.270833
86,6.170833,9.925000
87,6.662500,10.441667
88,6.883333,6.983333
89,5.662500,8.487500
90,10.412500,6.841667
91,8.720834,11.495833
92,16.325001,7.045833
93,5.145833,13.254167
94,7.683333,6.395833
95,6.183333,5.625000
96,5.704167,25.825001
97,9.845834,5.720833
98,10.770833,7.725000
99,8.062500,5.062500
100,7.904167,5.808333

Resultados Modelo 3: Período Fixo Aberto = 30

1,23.470833,29.237499
2,23.270834,26.691668
3,23.299999,28.104166
4,25.095833,25.408333
5,23.979166,35.779167
6,24.408333,23.741667
7,27.008333,24.195833
8,22.537500,25.266666
9,22.920834,25.062500
10,23.979166,23.470833
11,27.691668,23.150000
12,23.120832,27.900000
13,23.091667,29.704166
14,24.541666,21.116667
15,25.054167,24.337500
16,22.491667,26.495832
17,23.850000,23.125000
18,22.150000,25.299999
19,25.133333,26.762501
20,22.833334,22.691668
21,25.129168,27.345833
22,26.629168,26.058332
23,21.262501,22.725000
24,26.679167,20.641666
25,24.387501,28.133333
26,27.258333,24.079166
27,25.279167,26.991667
28,21.625000,23.933332
29,22.108334,25.804167
30,22.991667,21.833334
31,24.929167,24.612499
32,21.537500,25.729166
33,25.729166,27.166666
34,22.525000,25.491667
35,20.887501,25.004168
36,20.133333,22.645834
37,26.429167,24.595833
38,26.162500,23.525000
39,19.416666,23.758333
40,22.341667,21.049999
41,22.766666,19.679167
42,26.387501,27.341667
43,21.049999,24.616667
44,21.775000,20.866667
45,24.266666,27.799999
46,21.833334,30.154167
47,27.058332,24.891666
48,24.441668,28.866667
49,21.029167,26.920834
50,19.020834,26.358334

51,25.808332,22.341667
52,22.687500,20.541666
53,20.233334,25.679167
54,19.266666,23.054167
55,24.954166,25.570833
56,27.025000,25.337500
57,21.162500,26.833334
58,25.112499,26.775000
59,22.545834,21.674999
60,21.962500,22.616667
61,23.366667,26.129168
62,25.737499,24.379168
63,23.500000,19.004168
64,19.275000,28.104166
65,20.375000,23.670834
66,21.670834,22.400000
67,24.504168,27.962500
68,21.758333,24.891666
69,22.637501,28.508333
70,19.491667,22.016666
71,20.604166,24.270834
72,23.666666,27.020834
73,24.875000,24.437500
74,25.458334,26.674999
75,21.266666,24.325001
76,21.295834,24.200001
77,29.008333,20.279167
78,23.629168,29.841667
79,21.895834,25.245832
80,24.674999,21.720833
81,24.258333,25.350000
82,21.079166,23.066668
83,24.820833,28.825001
84,27.516666,24.116667
85,24.129168,24.616667
86,21.983334,28.712500
87,19.950001,27.375000
88,29.424999,22.770834
89,20.100000,22.854166
90,21.041666,28.970833
91,24.470833,22.841667
92,24.408333,23.295834
93,27.795834,21.195833
94,19.458334,23.820833
95,23.404167,26.875000
96,25.195833,22.612499
97,22.695833,23.304167
98,20.066668,25.491667
99,21.466667,20.762501
100,22.958334,24.337500

Resultados Modelo 3: Aberto A = 5, Aberto B = 5

1,6.345833,6.966667
2,8.287500,7.166667
3,8.112500,7.704167
4,7.033333,7.416667
5,6.354167,7.166667
6,9.137500,7.779167
7,6.966667,7.945833
8,6.633333,8.341666
9,8.766666,6.737500
10,6.941667,7.441667
11,8.816667,6.475000
12,6.554167,7.154167
13,6.037500,7.120833
14,9.279166,7.162500
15,7.337500,8.395833
16,7.950000,6.491667
17,8.312500,6.837500
18,6.870833,9.804167
19,5.587500,8.233334
20,5.916667,9.533334
21,7.245833,8.433333
22,6.166667,7.158333
23,8.020833,10.441667
24,6.325000,8.691667
25,6.395833,8.620833
26,6.945833,6.866667
27,6.783333,9.354167
28,6.562500,6.633333
29,5.970833,7.279167
30,7.495833,8.562500
31,7.025000,7.483333
32,8.329166,7.191667
33,8.125000,7.666667
34,5.966667,8.775000
35,8.300000,7.358333
36,6.450000,7.687500
37,8.704166,7.341667
38,9.320833,8.641666
39,7.558333,6.329167
40,7.133333,7.225000
41,5.675000,6.533333
42,8.104167,6.720833
43,6.875000,8.595834
44,7.179167,7.737500
45,9.345834,5.237500
46,9.612500,7.366667
47,6.770833,8.879167
48,6.166667,7.750000
49,10.445833,7.095833
50,6.583333,7.487500
51,11.229167,7.200000

52,8.241667,8.729167
53,7.550000,9.050000
54,12.250000,7.170833
55,7.466667,7.966667
56,6.875000,8.112500
57,6.704167,6.658333
58,8.395833,7.758333
59,6.591667,7.870833
60,6.275000,10.991667
61,7.062500,8.820833
62,7.487500,8.216666
63,7.962500,7.587500
64,6.704167,7.166667
65,6.979167,7.208333
66,7.729167,6.508333
67,7.062500,7.141667
68,7.625000,7.745833
69,9.337500,7.054167
70,5.650000,7.041667
71,6.304167,7.712500
72,8.820833,5.600000
73,7.220833,7.050000
74,8.754167,6.295833
75,5.637500,7.129167
76,7.433333,9.250000
77,6.566667,8.775000
78,7.304167,6.591667
79,7.225000,6.783333
80,8.412500,8.004167
81,6.204167,7.058333
82,7.216667,6.629167
83,6.225000,8.383333
84,12.404166,7.695833
85,7.662500,9.508333
86,6.945833,6.695833
87,5.908333,7.279167
88,7.075000,5.716667
89,7.020833,8.683333
90,7.216667,8.229167
91,6.683333,7.858333
92,6.620833,7.454167
93,9.158334,11.741667
94,7.316667,7.891667
95,6.937500,12.875000
96,7.441667,7.745833
97,6.687500,9.120833
98,6.458333,9.050000
99,6.804167,7.220833

100,6.808333,6.658333

Resultados Modelo 3: Aberto A = 15, Aberto B = 5

1,4.862500,57.158333
2,4.633333,65.454170
3,4.545833,87.920830
4,4.308333,85.537498
5,5.045833,51.504166
6,4.816667,57.191666
7,4.316667,55.174999
8,3.854167,33.162498
9,4.512500,60.799999
10,4.900000,45.179165
11,4.587500,38.033333
12,4.825000,51.345833
13,5.166667,43.724998
14,4.779167,80.275002
15,4.650000,62.508335
16,5.508333,59.779167
17,4.879167,69.449997
18,5.475000,70.074997
19,4.925000,59.458332
20,4.829167,90.991669
21,4.775000,88.970833
22,4.841667,46.216667
23,4.579167,68.629166
24,4.070833,84.079170
25,4.816667,46.608334
26,4.308333,58.150002
27,4.800000,50.229168
28,4.691667,60.866665
29,4.108333,72.883331
30,4.729167,66.208336
31,4.475000,64.941666
32,4.116667,45.424999
33,4.370833,61.129166
34,4.454167,30.141666
35,4.445833,60.908333
36,4.795833,69.979164
37,4.629167,59.866665
38,4.470833,37.345833
39,4.425000,81.724998
40,5.133333,31.066668
41,4.220833,26.195833
42,5.054167,47.845833
43,4.320833,106.449997
44,5.237500,87.795830
45,4.737500,69.308334
46,4.583333,62.299999
47,3.870833,69.820831
48,4.554167,39.395832
49,4.933333,62.166668
50,4.779167,83.283333

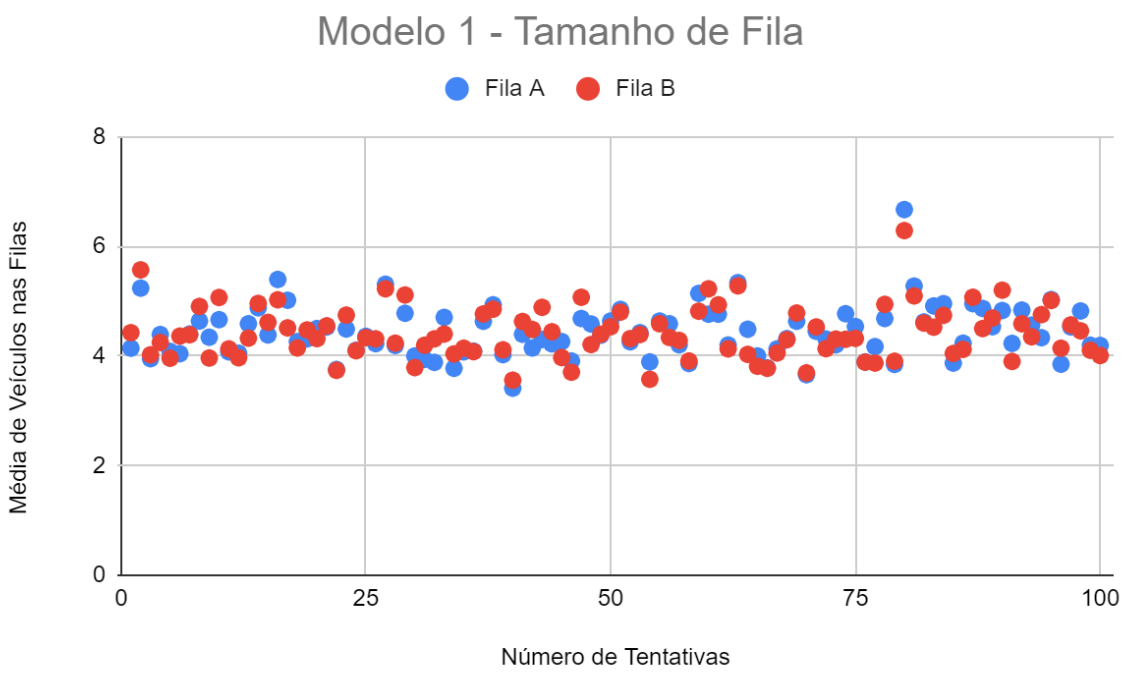
51,4.791667,70.845833
52,4.483333,76.500000
53,3.812500,51.112499
54,4.658333,39.700001
55,4.358333,53.258335
56,4.641667,92.962502
57,4.512500,53.387501
58,5.241667,79.008331
59,5.041667,74.066666
60,4.658333,44.566666
61,4.754167,44.945835
62,5.341667,43.558334
63,5.412500,37.362499
64,4.541667,77.175003
65,4.954167,87.045830
66,5.395833,55.150002
67,5.170833,51.387501
68,4.737500,76.675003
69,4.808333,38.408333
70,5.150000,52.637501
71,4.429167,62.575001
72,4.595833,69.554169
73,4.683333,58.387501
74,4.475000,27.100000
75,5.491667,34.391666
76,4.270833,112.908333
77,5.050000,73.733330
78,5.416667,91.616669
79,4.625000,61.554165
80,4.633333,37.941666
81,4.729167,49.891666
82,4.458333,34.754166
83,4.562500,63.270832
84,4.429167,76.716667
85,3.762500,59.295834
86,4.725000,34.912498
87,4.504167,71.241669
88,4.879167,44.612499
89,3.737500,107.945831
90,3.983333,114.020836
91,4.941667,109.512497
92,4.362500,88.758331
93,4.350000,66.079170
94,5.070833,38.429165
95,5.016667,54.029167
96,4.954167,68.387497
97,4.954167,84.033333
98,3.695833,85.154167
99,4.820833,64.583336
100,4.537500,57.329166

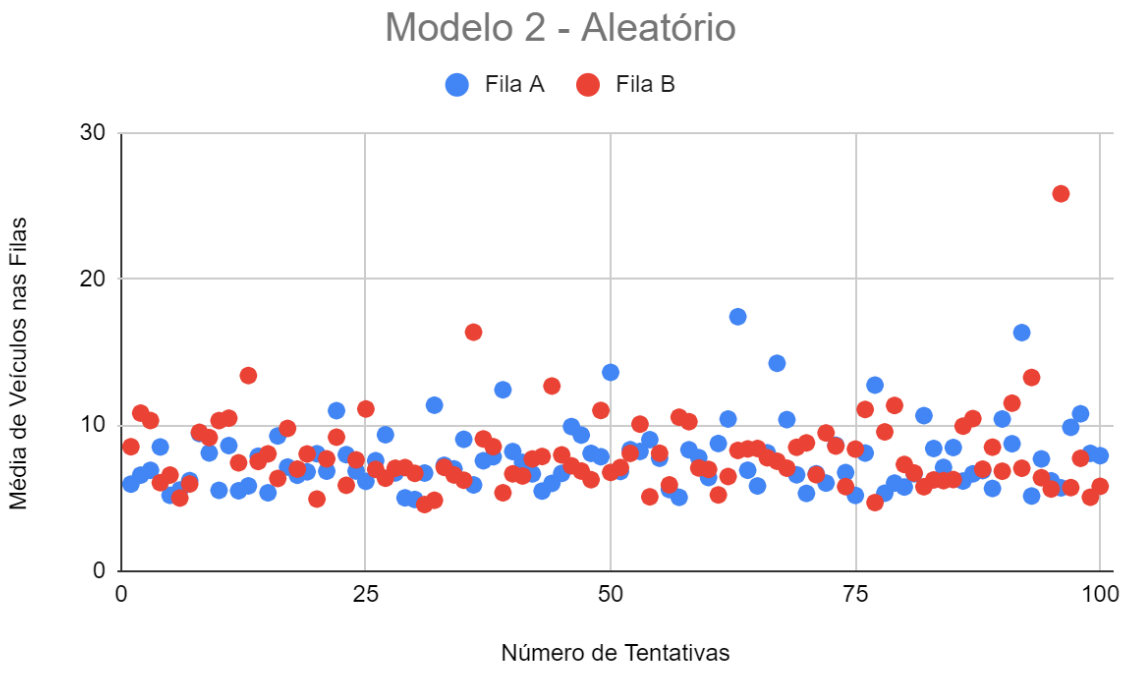
Resultados Modelo 3: Aberto A = 5, Aberto B = 15

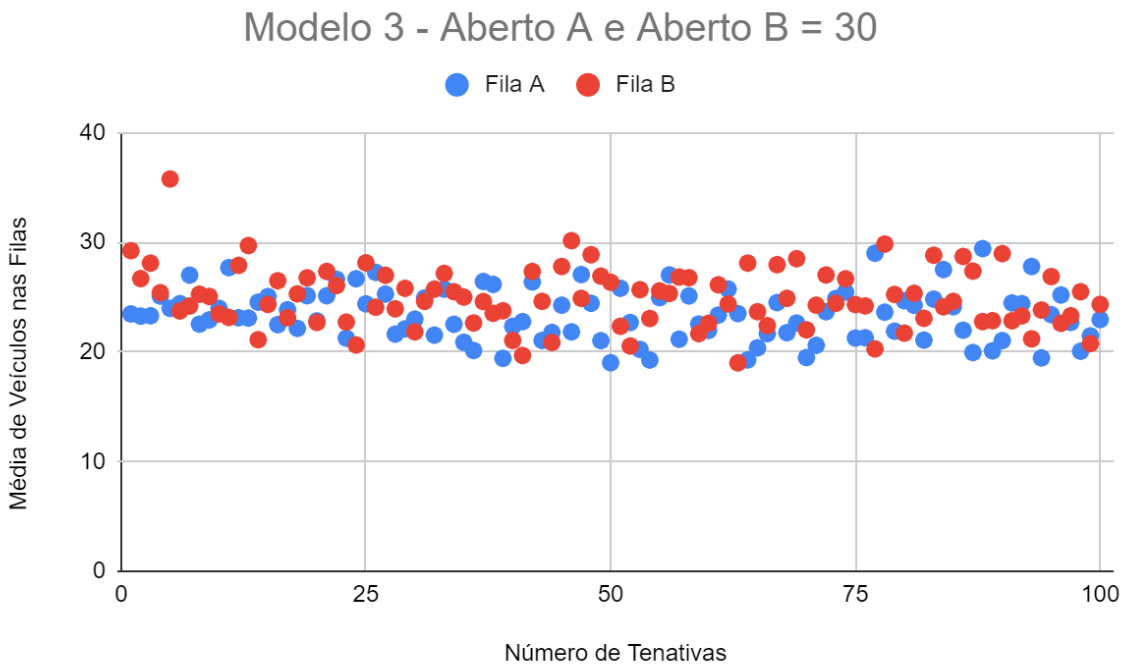
1,62.095833,4.212500
2,37.554165,4.750000
3,72.770836,4.562500
4,50.679165,4.275000
5,51.304165,4.612500
6,62.470833,5.391667
7,78.658333,4.287500
8,77.404167,4.833333
9,52.370834,4.400000
10,61.062500,5.000000
11,55.895832,5.645833
12,27.900000,5.200000
13,36.833332,4.666667
14,82.599998,5.075000
15,29.262501,4.391667
16,71.858330,4.675000
17,89.354164,5.870833
18,64.770836,4.225000
19,71.625000,4.679167
20,39.987499,4.054167
21,30.383333,4.604167
22,88.095833,4.425000
23,51.433334,5.120833
24,41.579166,4.550000
25,54.841667,3.883333
26,57.604168,4.712500
27,83.766670,4.112500
28,74.724998,5.716667
29,59.158333,4.741667
30,85.758331,4.566667
31,57.683334,4.441667
32,61.275002,4.137500
33,121.000000,4.125000
34,26.383333,4.900000
35,36.625000,4.908333
36,67.637497,5.491667
37,60.079166,4.854167
38,56.679165,5.625000
39,29.554167,4.458333
40,39.316666,5.141667
41,49.545834,4.829167
42,68.004166,4.525000
43,80.750000,4.779167
44,68.633331,5.379167
45,66.012497,4.954167
46,20.516666,4.666667
47,49.695835,5.054167
48,71.416664,4.941667

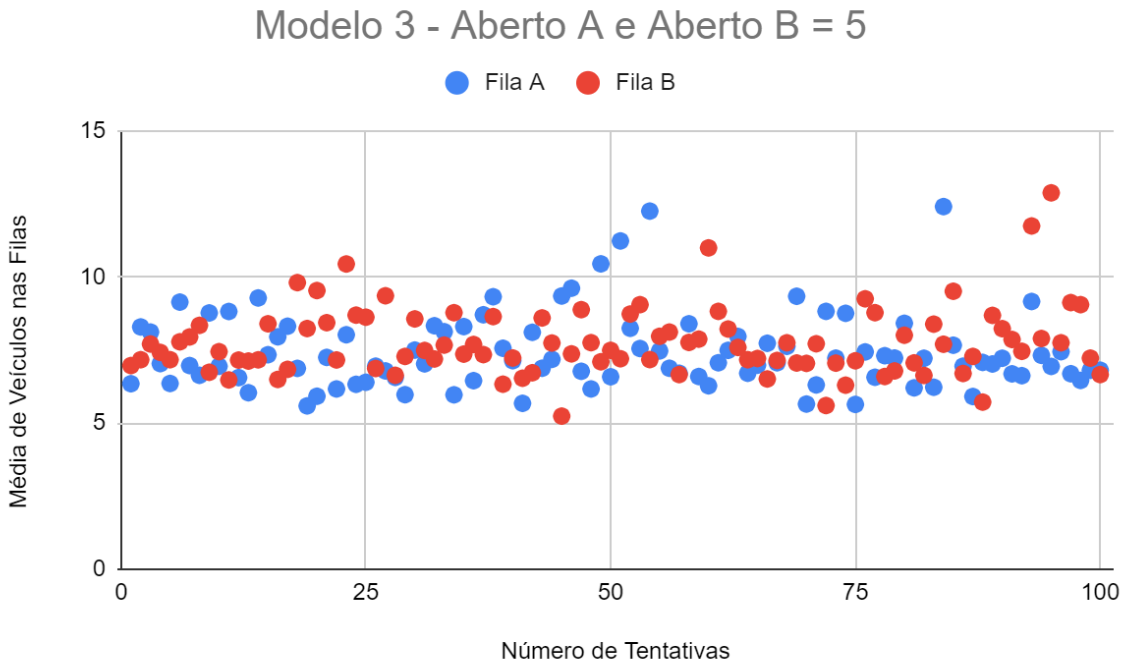
49,41.150002,4.345833
50,59.987499,5.137500
51,114.058334,5.141667
52,44.070835,4.966667
53,51.366665,5.162500
54,68.158333,5.029167
55,57.037498,5.229167
56,70.375000,5.208333
57,44.016666,4.454167
58,39.679165,3.812500
59,51.687500,4.495833
60,58.541668,5.170833
61,102.400002,5.158333
62,66.429169,4.654167
63,70.308334,4.245833
64,44.437500,3.954167
65,65.491669,5.000000
66,40.808334,4.679167
67,65.891670,4.241667
68,72.229164,4.004167
69,76.908333,4.529167
70,24.437500,5.029167
71,84.362503,4.108333
72,52.562500,4.333333
73,37.741665,5.166667
74,31.683332,4.837500
75,49.729168,4.241667
76,79.612503,4.654167
77,86.658333,4.412500
78,38.116665,5.333333
79,75.562500,5.495833
80,29.595833,4.520833
81,48.333332,4.333333
82,47.029167,5.104167
83,79.183334,5.050000
84,71.187500,4.950000
85,24.191668,5.283333
86,51.983334,4.712500
87,103.391670,4.050000
88,49.820835,4.420833
89,61.529167,4.745833
90,59.308334,4.870833
91,37.933334,4.404167
92,36.625000,5.295833
93,57.575001,5.633333
94,61.508335,5.141667
95,51.245834,4.833333
96,47.966667,4.883333
97,50.849998,4.075000
98,94.208336,4.466667
99,90.933334,4.912500
100,64.837502,4.879167

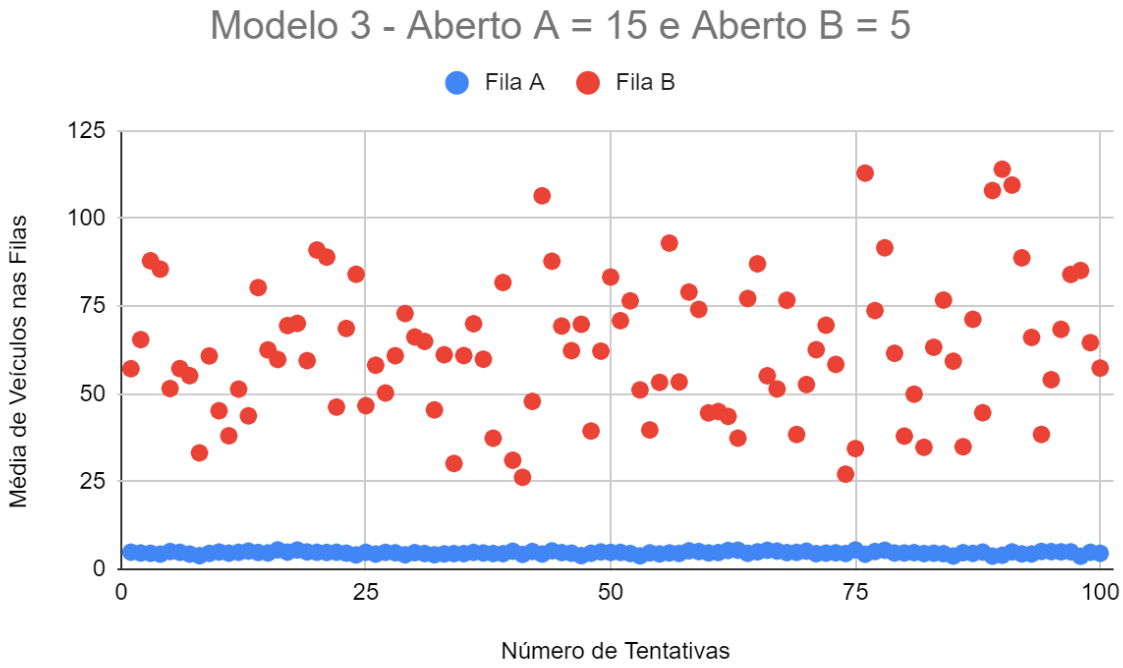
Analisando Resultados:

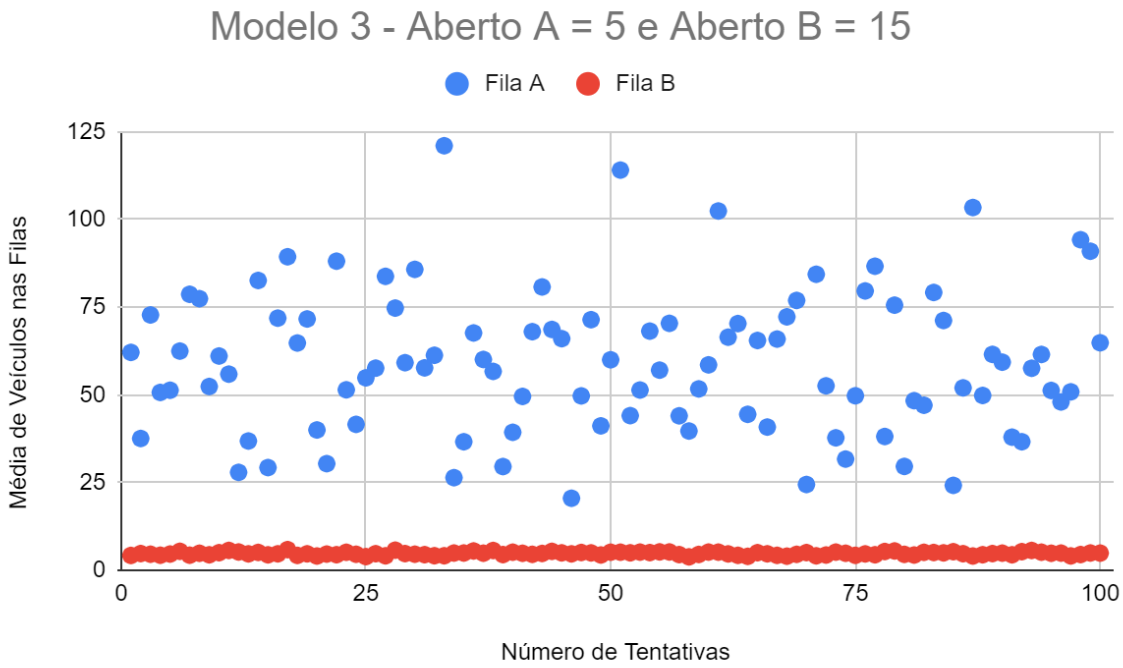


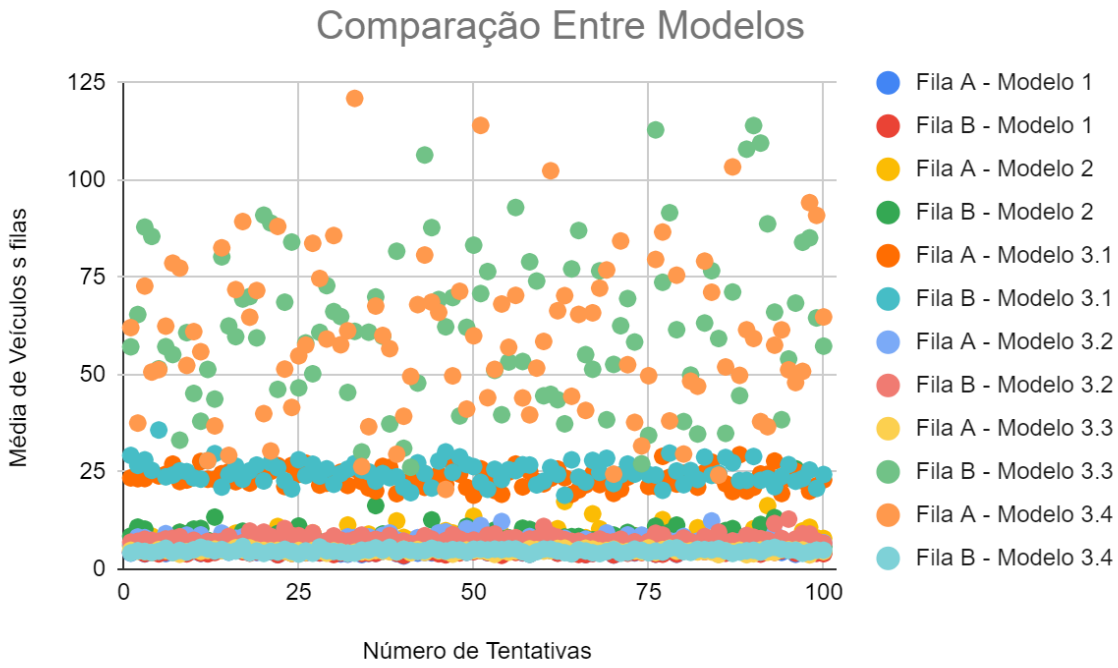



Modelo 3 - Aberto A e Aberto B = 30
Fila A
Fila B
Média de Veículos nas Filas
40
30
20
10
0
0
25
50
75
100
Número de Tenativas


Modelo 3 - Aberto A e Aberto B = 5
Fila A
Fila B
Média de Veículos nas Filas
15
10
5
0
0
25
50
75
100
Número de Tentativas


Modelo 3 - Aberto A = 15 e Aberto B = 5
Fila A
Fila B
125
100
75
50
25
0
0
25
50
75
100
Média de Veículos nas Filas
Número de Tentativas


Modelo 3 - Aberto A = 5 e Aberto B = 15
Fila A
Fila B
125
100
75
50
25
0
0
25
50
75
100
Média de Veículos nas Filas
Número de Tentativas



Foi realizada uma simulação com cada modelo desenvolvido a fim de verificar qual deles apresentaria um comportamento mais satisfatório para uma possível implementação do mesmo em uma situação de trânsito real. Os dados de taxa de entrada e saída das ruas são todos aleatórios. O modelo que apresentou o melhor desempenho foi aquele que adota a política de abertura e fechamento do semáforo de acordo com o tamanho das filas das ruas A e B (modelo 1). Esta simulação dos modelos foi pensada da seguinte maneira. O período total a ser verificado foi de 240 períodos de tempo e isto representa uma tentativa de teste do modelo em questão. Foram realizadas 100 tentativas de teste de cada modelo. Ou seja, foi simulado 100 vezes uma situação representada com duração de 240 períodos de tempo cada tentativa. Em relação aos outros modelos, foi constatado que o modelo com política de abertura e fechamento de maneira aleatória (modelo 2) não apresentou um resultado ruim, pois o número médio de veículos presente na fila não foi elevado, mas mesmo assim o modelo 1 apresentou uma média de veículos na fila inferior. Já no modelo com tempo fixo de aberto verificou-se que se adotado um tempo fixo de aberto de 1 período de tempo para a rua A e 1 período

de tempo para a rua B ele apresentou um resultado satisfatório pois foi melhor que o modelo 2 porém o modelo 1 prevalece na liderança. Já quando se aumenta o período de abertura de ambas as ruas para 5 períodos de tempo, a média de veículos nas filas aumenta consideravelmente. Já no último teste do modelo 3 a rua A permanece com o semáforo aberto por 15 períodos de tempo e o semáforo da rua B por 5 períodos de tempo. Há uma discrepância entre a média de veículos nas filas. A fila A com uma média de veículos nela satisfatória, porém na fila B o resultado não foi satisfatório, pois a média de veículos na sua fila foi elevada. Este modelo 3 não respondeu muito bem para valores maiores de tempo que os semáforos permanecerão abertos. Se ambos têm o mesmo valor de tempo aberto, mas se este tempo for mais elevado que 1 por exemplo, a tendência é que a média das filas também aumentem consideravelmente, por isso foi constatado que ele não é bom para a configuração de tempo aberto muito elevado. Ele funciona melhor com poucos períodos e se ambas as filas apresentarem o mesmo período aberto do controlador.

Conclusões:

De acordo com este estudo e com a implementação de um algoritmo próprio em linguagem C foi possível afirmar e concluir que o semáforo funcionado de acordo com o modelo 1 apresentou um resultado mais satisfatório se comparado com os outros modelos. Já que foram realizados testes simulando os modelos 100 vezes para averiguar as suas respostas. De acordo com este estudo é possível ter conclusões de que uma política de abertura e fechamento de semáforos faz toda a diferença no resultado final de um sistema. A escolha de uma política ineficaz pode levar a problemas sérios de mobilidade urbana. Foi um estudo meramente simulado com valores aleatórios, mas que pode sem sombras de dúvidas ser replicado numa situação de trânsito real se for dado continuidade

deste estudo iniciado na disciplina de Projeto Integrador III e que poderia fazer toda a diferença no trânsito de cidades como Florianópolis.

Trabalho Futuros:

Com este estudo realizado em menos de 6 meses de disciplina é possível dar continuidade em um projeto mais robusto que envolve uma cidade, uma prefeitura em específico. Para resolver o problema de trânsito de uma cidade, com a implementação de dispositivos controladores de trânsito inteligentes. A ideia da implementação de um dispositivo, de um sistema de trânsito com este teria que ter uma infraestrutura bem robusta para a sua real eficácia. A ideia deste sistema seria implementar um semáforo inteligente com a política presente no modelo 1, ou seja, o que vale é o maior tamanho de fila. Imagine a Beira Mar Norte em Florianópolis, lá há diversos semáforos. A ideia é todos aqueles semáforos conseguirem se comunicar entre si através da internet. Então entram conceitos de IoT (Internet of things) nesta aplicação. Haveria câmeras no ambiente, representando aquele ambiente de observação nos algoritmos. Estas câmeras seriam responsáveis de observar o ambiente e passar para o controlador de trânsito qual semáforo ele deveria abrir. Todas as decisões seriam em tempo real de acordo com o fluxo do momento e não de uma média averiguada em um estudo, por exemplo. Os semáforos funcionam de maneira dinâmica. Eles também interagiriam com veículos, podendo verificar o tempo de reação dos motoristas. Não é possível encontrar um semáforo inteligente como este em Florianópolis. Semáforos inteligentes seriam a solução para problemas de trânsito na cidade, mas é preciso muito investimento e algum investidor interessado na ideia e colocá-la em prática. Esta ideia é provável que exista no mundo da IoT mas não é encontrada em Florianópolis. Basta apenas andar de carro pela cidade e verificar o funcionamento dos semáforos. Vai encontrar situações em que há semáforo aberto para uma fila vazia e fechado para um fluxo intenso de veículos. Isto acontece pois os semáforos não são inteligentes e o que prejudica muito a mobilidade urbana de Florianópolis, já que a mesma é uma ilha. A implementação de semáforos inteligentes minimizaria um pouco o trânsito já que a mobilidade não

depende só de controladores e sim da infraestrutura urbana. Se não há muito planejamento urbano, não é possível minimizar todo este problema de mobilidade urbana em Florianópolis. Outro fator que poderia ajudar semáforos inteligentes seria os veículos autônomos, pois permitiriam uma interação via internet com o controlador de trânsito. Quem quiser dar continuidade neste projeto há muito futuro.

Referências

https://queue-it.com/blog/queuing-theory/

ALSABAAN, Maazen et al. **Vehicular Networks for a Greener Environment**: A Survey. IEEE, 2013.

BORGES, Dimitrius F. et al. **Traffic Light Control Using Hierarchical Reinforcement Learning and Options Framework,** Minas Gerais: IEEE, 2021.

BOUKERCHE, Azzedine e YOUNES, Maram B. **Intelligent Traffic Light Controlling Algorithms Using Vehicular Networks.** IEEE, 2016.

Queuing theory: Definition, history & real-life applications. Disponível em: https://queue-it.com/blog/queuing-theory/. Acesso em: 3 de mar. 2022.