iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

TG I Modelação Estocástica

Simulação da Gestão de Contactos

André Silvestre Nº100122
Eliane Gabriel Nº102303
Maria João Loureiro Nº106748
Margarida Pereira Nº105677
Umaima Mahomed Nº 95229

10 de novembro de 2023

- **Ano Letivo:** 2023/24
- **Docentes:** Catarina Marques e Pedro Serrasqueiro

Gestão dos Contactos Pré-Operatórios

Serviço de Cirurgia de um Hospital

O **objectivo** deste estudo é optimizar a gestão dos contactos pré-operatórios, minimizando o número de administrativos afectos a este processo. Para tal, pretende-se que simule a gestão dos contactos de forma a analisar o impacto da ocorrência de diferentes cenários na eficiência do processo de gestão de contactos (tempo total do processo e total de chamadas perdidas).

Características do processo de gestão de contactos – Situação Actual:

1. Existem apenas **2 administrativos** no serviço, que começam a trabalhar às $8h$ e terminam às $18h$ de cada dia útil.
2. O tempo entre chegadas de chamadas ou mensagens tem distribuição $Exp(\frac{1}{\lambda})$, sendo a taxa de contacto (λ) de 40 chamadas ou mensagens por hora, o equivalente a $\frac{40}{60} = \frac{1}{1.5}$ (por minuto).
3. Os pacientes entram em contacto por telefone com probabilidade 0.85 ou por via eletrónica com probabilidade 0.15.
4. Relativamente ao agendamento de consultas,
   - 30% dos vezes não é agendada uma consulta;
   - 70% das vezes é agendada uma consulta, das quais:
     - $\frac{2}{3}$ das vezes a consulta ocorre por telefone e;
     - $\frac{1}{3}$ é presencial.
5. O tempo necessário que os administrativos necessitam para fazer as perguntas de triagem segue distribuição $N(\mu = 4, \sigma = 1)$, enquanto o tempo de tomada de decisão sobre qual consulta marcar, assim como, o tempo de agendamento da consulta (se necessário) seguem ambos distribuição $N(\mu = 1, \sigma = 0.25^{2})$.
6. O sistema telefónico instalado designa uma chamada após 5 minutos de espera para ser atendida. Nesta situação, a chamada é perdida e o paciente terá de entrar em contacto de novo.

(1) Note-se que estes tempos são sempre maiores ou iguais a zero pelo que deve ser usada a distribuição normal truncada a valores não negativos.

```
# Remover todos os objetos da área de trabalho.
rm(list=ls(all=TRUE))

# Bibliotecas utilizadas
library(timer)
library(timer-plot)
library(reshape2)     # Para transformar o df resultante de 'get_mon_attributes()'
library(flextable)    # Tabela
library(truncnorm)    # Truncar os valores da Normal
require(ggflow)

# Defini a semente para reprodutibilidade dos resultados
set.seed(2023)
```

Processo Atual

```
##########################  Métricas a analisar  ##########################
metricas_func = function(gestao_contactos) {

  contactos = get_mon_arrivals(gestao_contactos, ongoing=TRUE) %>%
    transform(waiting_time = end_time - start_time - activity_time)

  # uniformizar dos ms (caso em que não houve atendimento)
  contactos$activity_time = ifelse(contactos$activity_time == 0, NA,contactos$activity_time)

  # disol / fonte: https://www.rdocumentation.org/packages/reshape2/versions/1.4.4/topics/cast
  contactos_atributos <- dcast(gestao_contactos %>% get_mon_attributes(),
                               replication + name ~ key,
                               value.var = "value")

  # Juntar todas as informações da simulação num só data.frame
  contactos_v = merge(contactos, contactos_atributos, by = c("name", "replication"), all = TRUE)

  # Número de Pacientes (Chamadas + Mensagens)
  pacientes_total = nrow(contactos_v,max(contactos$replication)

  # Número de Pacientes Atendidos (Chamadas + Mensagens)
  pacientes_total_atendidos = sum(!is.na(contactos$activity_time ))/max(contactos$replication)

  # Número de Pacientes Não Atendidos (Chamadas + Mensagens)
  pacientes_total_nao_atendidos = sum(is.na(contactos$activity_time ))/max(contactos$replication)

  # Número de chamadas telefónicas atendidas
  chamadas_atendidas = sum(contactos$Chamada == 1 &
                           !is.na(contactos$activity_time), na.rm = TRUE)/max(contactos$replication)

  # Número de chamadas perdidas
  chamadas_perdidas = sum(contactos$Chamada == 1 &
                          is.na(contactos$activity_time), na.rm = TRUE)/max(contactos$replication)

  # Número de mensagens respondidas e não respondidas
  mensagens_respondidas = sum(contactos$Mensagem == 1 &
                              !is.na(contactos$activity_time ), na.rm = TRUE)/max(contactos$replication)

  mensagens_nao_respondidas = sum(contactos$Mensagem == 1 &
                                  is.na(contactos$activity_time ), na.rm = TRUE)/max(contactos$replication)

  # Tempo de espera médio
  tempos_medios <- numeric(max(contactos$replication))
  for (i in 1:max(contactos$replication)){
    replica <- subset(contactos, replication == i$activity_time
    tempos_medios[i] <- mean(replica, na.rm = T)
  }
  tempo_espera_medio = mean(tempos_medios)
  sd_tempo_espera_medio = ifelse(length(tempos_medios) > 1, sd(tempos_medios), sd(replica, na.rm = T))

  # Tempo de serviço
  tempos_servico <- numeric(max(contactos$replication))
  for (i in 1:max(contactos$replication)){
    replica <- subset(contactos, replication == i$activity_time
    tempos_servico[i] <- mean(replica, na.rm = T)
  }
  tempo_servico_medio = mean(tempos_servico)
  sd_tempo_servico_medio = ifelse(max(contactos$replication) > 1, sd(tempos_servico), sd(replica, na.rm = T))

  # Tempo total de processo (tempo médio de atendimento)
  tempos_total <- numeric(max(contactos$replication))
  for (i in 1:max(contactos$replication)){
    replica <- subset(contactos, replication == i)$end_time - replica$start_time, na.rm = T)
    tempos_total[i] <- mean(replica$end_time - replica$start_time, na.rm = T)
  }
  tempo_total_processo = mean(tempos_total)
  sd_tempo_total_processo = ifelse(length(tempos_total) > 1,
                                   sd(tempos_total), sd(replica$end_time - replica$start_time), na.rm = T))

  # Tempo de espera médio (Chamadas)
  tempos_medios_chamadas <- numeric(max(contactos$replication))
  for (i in 1:max(contactos$replication)){
    replica <- subset(contactos, replication == i & Chamada == 1)$activity_time
    tempos_medios_chamadas[i] <- mean(replica, na.rm = T)
  }
  tempo_espera_medio_chamadas = mean(tempos_medios_chamadas)
  sd_tempo_espera_medio_chamadas = ifelse(length(tempos_medios_chamadas) > 1,
                                          sd(tempos_medios_chamadas), sd(replica, na.rm = T))

  # Tempo de espera médio (Mensagens)
  tempos_medios_mensagens <- numeric(max(contactos$replication))
  for (i in 1:max(contactos$replication)){
    replica <- subset(contactos, replication == i & Mensagem == 1)$activity_time
    tempos_medios_mensagens[i] <- mean(replica, na.rm = T)
  }
  tempo_espera_medio_mensagens = mean(tempos_medios_mensagens)
  sd_tempo_espera_medio_mensagens = ifelse(length(tempos_medios_mensagens) > 1,
                                           sd(tempos_medios_mensagens), sd(replica, na.rm = T))

  # Número de pacientes com consulta presencial:
  pacientes_consulta_presencial = sum(!is.na(contactos$CAP))/max(contactos$replication)

  # Número de pacientes com consulta telefónica
  pacientes_consulta_telefonica = sum(!is.na(contactos$CAT))/max(contactos$replication)

  # Número de pacientes sem necessidade de consulta
  pacientes_sem_necessidade = sum(!is.na(contactos$CNA))/max(contactos$replication)

  # Crie um data frame com as métricas
  nomes_metricas <- c("Total de Pacientes", "Total de Pacientes Atendidos", "Pacientes Totais Não Atendidos",
                      "Total de Chamadas","Chamadas Telefónicas Atendidas","Chamadas Perdidas",
                      "Total de Mensagens", "Mensagens Respondidas", "Mensagens Não Respondidas",
                      "Pacientes Consulta Presencial", "Pacientes Consulta Telefónica",
                      "Pacientes Sem Necessidade", "Tempo Médio dos Pacientes atendidos (minutos)",
                      "Tempo de Espera Médio", "Tempo de Espera Médio (Chamadas)", "Tempo de Espera Médio (Mensagens)",
                      "Tempo de Serviço Médio", "Tempo Total de Processo")

  output <- data.frame(
    Métrica = nomes_metricas,
    Valor = c(
      round(pacientes_total),
      round(pacientes_total_atendidos),
      round(pacientes_total_nao_atendidos),
      round(chamadas_atendidas + chamadas_perdidas),
      round(chamadas_atendidas),
      round(chamadas_perdidas),
      round(mensagens_respondidas + mensagens_nao_respondidas),
      round(mensagens_respondidas),
      round(mensagens_nao_respondidas),
      round(pacientes_consulta_presencial),
      round(pacientes_consulta_telefonica),
      round(pacientes_sem_necessidade), "Média",
      formatC(round(tempo_espera_medio,2),1,format="f"),
      formatC(round(tempo_espera_medio_chamadas,2),1,format="f"),
      formatC(round(tempo_espera_medio_mensagens,2),1,format="f"),
      formatC(round(tempo_servico_medio,2),1,format="f"),
      formatC(round(tempo_total_processo,2),1,format="f"),
    Percentagem = c("",
      formatC(round(pacientes_total_atendidos / pacientes_total * 100, 1),1,format="f"),
      formatC(round(pacientes_total_nao_atendidos / pacientes_total * 100, 1),1,format="f"), "",
      formatC(round(chamadas_atendidas/(chamadas_atendidas + chamadas_perdidas) * 100, 1),1,format="f"),
      formatC(round(chamadas_perdidas/(chamadas_atendidas + chamadas_perdidas) * 100, 1),1,format="f"), "",
      formatC(round(mensagens_respondidas / mensagens * 100, 1),1,format="f"), formatC(round(mensagens_nao_respondidas / mensagens * 100, 1),1,format
="f"),
      formatC(round(pacientes_consulta_presencial / pacientes_total_atendidos * 100, 1),1,format="f"),
      formatC(round(pacientes_consulta_telefonica / pacientes_total_atendidos * 100, 1),1,format="f"),
      formatC(round(pacientes_sem_necessidade / pacientes_total_atendidos * 100, 1),1,format="f"), "DP",
      formatC(round(sd_tempo_espera_medio, 1),1,format="f"),
      formatC(round(sd_tempo_espera_medio_chamadas, 1),1,format="f"),
      formatC(round(sd_tempo_espera_medio_mensagens, 1),1,format="f"),
      formatC(round(sd_tempo_servico_medio, 1),1,format="f"),
      formatC(round(sd_tempo_total_processo, 1),1,format="f"))
  )

  colnames(output) <- c("Métrica", "Valor", "%")

  return(output)
}
```

```
##########################  Exercício 1  ##########################

# Realizar a simulação do processo de gestão de contactos no período das 8h às 18h para um ano.
# 1 / Analise o impacto do processo actual. Use 1 a 50 réplicas. Identifique possíveis ações de melhoria.

# Defini o tempo de contacto
taxa_contacto <- 40 / 60          # Converter a taxa para contactos por minuto

# Defini as distribuições do tempo de serviço
# Nota: Para cumprir com a atenção referida no enunciado de que os tempos são sempre >= 0, de modo a
#       usar a distribuição normal truncada a valores negativos, optámos por a biblioteca 'truncnorm'
# Documentação consultada: https://cran.r-project.org/web/packages/truncnorm/truncnorm.pdf
tempo_triagem <- function() rtruncnorm(1, a=0, b=Inf, mean = 4, sd = 1)
tempo_decisao <- function() rtruncnorm(1, a=0, b=Inf, mean = 1, sd = 0.25)
tempo_agendamento <- function() rtruncnorm(1, a=0, b=Inf, mean = 1, sd = 0.25)
```

Trajetórias dos Contactos

Os pacientes podem contactar o hospital de duas formas: por **Chamada Telefónica** ou via **Mensagem na App**, assim foram definidas 2 trajetórias possíveis.

Na 1ª trajetória, quando um paciente faz uma Chamada Telefónica, o sistema:

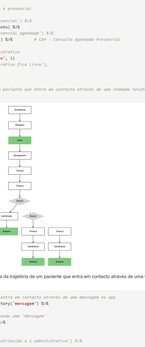
1. Identifica o contacto como uma `"Chamada"` e atribui um administrativo.
2. Verifica se a chamada é perdida após 5 minutos de espera.
3. Se o administrativo fica disponível antes da chamada ser perdida, o abandono é cancelado.
4. Realiza a triagem do paciente.
5. Tomada de decisão sobre a necessidade de consulta.
6. Se necessário, agendamento da consulta, por telefone ou presencial.

Já na 2ª trajetória, se o paciente enviar uma Mensagem na App, o sistema:

1. Identifica o contacto como uma `"Mensagem"` e atribui um administrativo.
2. Realiza a triagem do paciente.
3. Tomada de decisão sobre a necessidade de consulta.
4. Se necessário, agendamento da consulta, por telefone ou presencial.

Ás notas que ambas as trajetórias incorporam todos os processos através de bifurcações utilizando a função `branch()`: tendo por base as distribuições e probabilidades dadas.

```
# Trajetória de um paciente que entra em contacto através de uma chamada telefónica
contacto_pacientes_tel <- trajectory("chamada") %>%

  # Identifica o contacto como sendo uma 'Chamada'
  set_attribute("Chamada", 1) %>%

  # O sistema telefónico bloqueiga depois uma chamada após 5 minutos de espera para ser atendida.
  renege_in(t = 5, out = trajectory(
                                    # Ok log("Chamada perdida após 5 minutos")
                                    ) %>%

  seize("administrativo") %>%
    # log("Chamada / Paciente atribuída a 1 administrativo") %>%

  # Se o administrativo ficar livre antes da chamada do paciente ser perdida, o abandono deve ser cancelado
  renege_abort() %>%

  # Triagem
  timeout(tempo_triagem) %>%
    # log("Triagem concluída") %>%

  # Tomada de decisão
  timeout(tempo_decisao) %>%
    # log("Decisão tomada") %>%

  # Agendamento de consulta (caso haja necessidade)
  branch(
    function() {runif(1) > 0.3} * 1, continue = c(FALSE,FALSE),

    # Trajetória caso a consulta não seja agendada (30%)
    trajectory() %>%
      # log("Consulta não agendada") %>%
    set_attribute("CNA", 1) %>%                    # CNA - Consulta Não Agendada

      # Liberação do administrativo
    release("administrativo", 1),
      # %>% log("Administrativo fica livre"),

    # Trajetória caso a consulta seja agendada (70%)
    trajectory() %>%
      # log("Consulta a agendar") %>%

    branch(
      function() {runif(1) > 0.67} * 1, continue = c(FALSE,FALSE),

      # Caso em que a consulta é por telefone
      trajectory() %>%
        # log("Consulta por telefone") %>%
      timeout(tempo_agendamento) %>%
        # log("Consulta por telefone agendada") %>%
      set_attribute("CAT", 1) %>%                  # CAT - Consulta Agendada por Telefone

      # Liberação do administrativo
      release("administrativo",1),
        #  %>% log("Administrativo fica livre")

      # Caso em que a consulta é presencial
      trajectory() %>%
        # log("Consulta presencial") %>%
      timeout(tempo_agendamento) %>%
        # log("Consulta presencial agendada") %>%
      set_attribute("CAP", 1) %>%                  # CAP - Consulta Agendada Presencial

      # Liberação do administrativo
      release("administrativo", 1)
        # %>% log("Administrativo fica livre"),
    )
  )

# Fluxograma da trajetória de um paciente que entra em contacto através de uma chamada telefónica
plot(contacto_pacientes_tel)
```


Seizeadministrativo
Renege
Renege
Timeout
Timeout
Timeout
Release
Seizeadministrativo
Timeout
Timeout
Timeout
Seizeadministrativo
Timeout
Timeout
Timeout
Seizeadministrativo
Timeout
Timeout
Timeout

```
  set.attributes("CNA", 1) %>%

  # Liberação de administrativo
  release("administrative", 1),
    # %>% log_("Administrative free time"),

# Trajetória caso o consulte seja agendado (MR)
trajectory() %>%

  branch(
    function(): (runif(1) > 0.629.7) + 1, continue = c(FALSE,FALSE),

    # Caso em que a consulta é por telefone
    trajectory() %>%
      # log_("Consulta por telefone a agendar") %>%
      timeout(tempo_agendamento) %>%
      # log_("Chamada por telefone agendada") %>%
      set_attributes("CAT", 2) %>%                # Cat - Consulta Agendada por Telefone

      # Liberação de administrativo
      release("administrative", 1),
      # %>% log_("Administrative free time")

    # Caso em que a consulta é presencial
    trajectory() %>%
      # log_("Consulta presencial a agendar") %>%
      timeout(tempo_agendamento) %>%
        # log_("Consulta presencial agendada") %>%
      set_attributes("CAT", 1) %>%                # Cat - Consulta Agendada Presencial

      # Liberação de administrativo
      release("administrative", 1)
        # %>% log_("Administrative free time"),
    )
)

# Fluxograma do trajetória de um paciente que entra em contato através de uma mensagem na app
plot(contacts_pacientes_sms)
```


Figura 2 - Fluxograma da trajetória de um paciente que entra em contato através de uma **Mensagem na app**.

Situação Atual I 1 Réplica

O estudo do serviço de cirurgia do hospital na situação atual envolveu a simulação singular do seu funcionamento num período de 5 dias úteis, das 8h as 18h, com dois administrativos disponíveis. Neste cenário, os pacientes entraram em contato através de chamadas telefônicas (65%), mensagens (15%) ou pessoalmente ($\hat{E}xp(\lambda = 40)$).

Em comparação com o funcionamento do algoritmo implementado e para melhor compreender os atuais problemas e evolução do sistema, utilizaram-se os `log_` que foram posteriormente comentados no código.

```
set.seed(2023)

administration_schedule_1 <- schedule(timetable = c(0*60,10*60), values = -c(2, 0), period = 24*60)

# Trajetória para escolher o tipo de contato com base nas probabilidades
tipo_contato <- trajectory("Tipo de Contato") %>%
                branch(function(): (runif(1) > 0.65) + 1, continue = c(FALSE, FALSE),
                       # Trajetória caso o contacto tenha sido por telefone
                       contacts_pacientes_tel,

                       # Trajetória caso o contacto tenha sido por via eletrónica
                       contacts_pacientes_sms)

# Criar um ambiente de simulação, replicando o sistema apenas 1 vez da Situação Atual (SA)
gestao_contactos_SA_1 <- simmer("Gestao dos Contactos") %>%
  add_resource("administrative", administration_schedule_1)

# Gerar Pacientes no Período dos 8h às 18h em 5 dias
# Utilizamos a chaz.7ter para repetir a simulação dos pacientes nos 5 dias
# chaz: https://github.com/r-simmer/simmer.plot/blob/7244b92cf0e0ba05a8d22409b1239f8972335/R/plot.arrivals.R
for(i in 1:5) gestao_contactos_SA_1 %>%
  add_generator(paste("Paciente",i),
                trajectory = tipo_contato,
                from_to(start_time = ((i-1)*24*60)+8*60,
                        stop_time = ((i-1)*24*60)+18*60,
                        dist = function() c(0,rexp(10*10, rate = taxa_contacto),-1)
                        ),
                mon = 2 # Se o simulador deve monitorar as chegadas geradas ou não :
                        # 0 = não monitorizado
                        # 1 = monitorizando de chegada simples,
                        # 2 = nível 1 = monitorizando de atributo de chegada
                        )

  gestao_contactos_SA_1 %>% run(until = 5 * 24 * 60)  # Minutos de Simulação = 5 dias x 24 horas/dia x 60 minutos

## simmer environment: gestao dos Contactos | now: 7200 | next: 7680
## {Monitor: in memory }
## {Resource: administrative | monitored: TRUE | server status: 0(0) | queue status: 0(Inf) }
## {Source: Paciente 1 | monitored: 2 | n_generated: 462 }
## {Source: Paciente 2 | monitored: 2 | n_generated: 378 }
## {Source: Paciente 3 | monitored: 2 | n_generated: 392 }
## {Source: Paciente 4 | monitored: 2 | n_generated: 406 }
## {Source: Paciente 5 | monitored: 2 | n_generated: 347 }


# Métricas para a análise de eficiência
df <- metrics4.fun(gestao_contactos_SA_1)
ftable_1 <- ftable(df)

ftable_1 <- ftable(ftable_1, bg = "#0D2B6", part = "header")
ftable_1 <- color(ftable_1, color = "white", part = "header")
ftable_1 <- bold(ftable_1, bold = TRUE, part="header")
ftable_1 <- autofit(ftable_1)
ftable_1 %>%
  bold(part = "body", j= 1, j = 1) %>%
  hline(i = 1, part = "body", border = fp_border(color = "#002060", style = "dashed", width = 1)) %>%
  bold(part = "body", j= 4, j = 4) %>%
  hline(i = 4, part = "body", border = fp_border(color = "#002060", style = "dashed", width = 1)) %>%
  bold(part = "body", j= 5, j = 5) %>%
  hline(i = 5, part = "body", border = fp_border(color = "#002060", style = "dashed", width = 1)) %>%
  bold(part = "body", j= 11, j = 11) %>%
  hline(i = 12, part = "body", border = fp_border(color = "#002060", style = "solid", width = 1))
```

Métrica	Valor	%
Total de Pacientes	1961	
Total de Pacientes Atendidos	1013	51.7
Pacientes Totais Não Atendidos	948	48.3
Total de Chamadas	1675	
Chamadas Telefónicas Atendidas	727	43.4
Chamadas Perdidas	948	56.6
Total de Mensagens	286	
Mensagens Respondidas	286	100.0
Mensagens Não Respondidas	0	0.0
Pacientes Consulta Presencial	97	9.6
Pacientes Consulta Telefónica	628	62.0
Pacientes Sem Necessidade	288	28.4
Tempo Médio dos Pacientes Atendidos (minutos)	**Média**	**DP**
Tempo de Espera Médio	5.25	27.0
Tempo de Espera Médio [Chamadas]	4.25	1.3
Tempo de Espera Médio [Mensagens]	11.15	70.3
Tempo de Serviço Médio	5.71	1.1
Tempo Total de Processo	8.21	27.2

Tabela 1 - Variáveis de análise da eficiência da 1ª Simulação - **Situação Atual** I 1 Réplica.

Através da análise dos resultados (**Tabela 1**), numa simulação de 1961 pacientes, revelou-se que 48% dos pacientes não foram atendidos devido ao tempo de espera da chamada, sendo que nenhuma mensagem ficou por ser respondida.

```
# Representações gráficas da simulação
gestao_contactos_SA_1 %>% get_mon_resources() %>% plot(metric = "utilization")
gestao_contactos_SA_1 %>% get_mon_resources() %>% plot(metric = "usage", steps=T)
```

Figura 3 e 4 - Gráficos que representa o quanto um recurso (os 2 administrativos) é utilizado (esquerda); quando os recursos (administrativos) estão ocupados, bem como quantas pacientes estão na fila e como evoluem ao longo do tempo (**Situação Atual** em 1 réplica.

- A linha verde representa o número de administrativos ocupados; em todo o intervalo de simulação, ambos os administrativos estão ocupados.
- A linha vermelho representa o número de pacientes na fila que estão a espera para serem atendidas.
- A linha azul é o número total de pacientes no sistema: os que estão na fila mais os que estão a ser atendidos.

Relativamente aos recursos, os dois administrativos estão a ser utilizados a cerca de 96%, pelo que estes podem causar algum estrangulamento ao sistema (**Figura 3**).

Analisando os tempos médios dos pacientes atendidos, em minutos, o tempo total do processo foi de 8.2, com um tempo de espera, em média, de 5.3 e de serviço de 5.7.

Através da **Figura 4**, é evidente que os administrativos estão a ser utilizados em todo o período de simulação e que existem sempre pacientes a serem atendidos ou a espera de o serem. Os casos assinalados a azul que transpassam o um dia laboral para outro são referentes a mensagens que ficaram pendentes, contudo, estas foram resolvidas no dia seguinte, estando no final do tempo de simulação 100% das mensagens respondidas.

```
# De forma a tratar primeiro a 'waiting_time' antes de a visualizar graficamente, optamos por fazer o gráfico de raiz, utili
zando o código da biblioteca no mesmo
# Fonte: https://github.com/r-simmer/simmer.plot/blob/7244b92cf0e0ba05a8d22409b1239f8972335/R/plot.arrivals.R
gestao_contactos_SA_1_filt <- gestao_contactos_SA_1 %>%
  get_mon_arrivals() %>%
  dplyr::mutate(waiting_time = end_time - start_time - activity_time) %>%
  # Como os messages podem não terminar num si dia, nos casos em que já estava no final de forma das 18h, retire-se a hora
 i em espera
  # entre as 18h e as 8h do dia seguinte (equivale a 2anoh = 840 minutos) para melhor visualização dado serem poucos casos
  transform(waiting_time = ifelse(waiting_time > 840, waiting_time - 840, waiting_time))

ggplot(gestao_contactos_SA_1_filt) +
  ecdf(x = gestao_contactos_SA_1_filt$end_time, y = gestao_contactos_SA_1_filt$waiting_time) +
  geom_line(aes(group = gestao_contactos_SA_1_filt$replication),
            alpha = 1.0 / (log(max(gestao_contactos_SA_1_filt$replication)) + 1)) +
  stat_smooth() +
  xlab("simulation time") +
  ylab("waiting time") +
  ggtitle("Waiting time evolution") +
  expand_limits(y = 0)

gestao_contactos_SA_1 %>% get_mon_arrivals() %>% plot(metric = "activity_time")
```

Figura 5 e 6 - Gráficos que representam quanto tempo os pacientes estão na fila e o tempo do seu atendimento - **Situação Atual** em 1 réplica.

As **Figuras 5 e 6** revelam que os pacientes aguardam, em média, cerca de 5 minutos, podendo chegar a 10 minutos em alguns casos, enquanto o tempo de atividade geralmente se estende por 5 minutos ou mais.

Situação Atual I 50 Réplicas

```
set.seed(2023)

# Criar um ambiente de simulação, replicando o sistema 50 vezes da Situação Atual
administrations_schedule_1 <- schedule(timetable = c(0*60,10*60), values = -c(2, 0), period = 24*60)

# Criar um ambiente de simulação, replicando o sistema 50 vezes da Situação Atual
gestao_contactos_SA_50 <-
  lapply(1:50, function(i){
    gestao_contactos_SA <- simmer(paste("Gestao dos Contactos 50rep", i)) %>%
    add_resource("administrative", administrations_schedule_1)

    for(i in 1:5) gestao_contactos_SA %>%
    add_generator(paste("Paciente",i,i),
                  trajectory = tipo_contacto,
                  from_to(start_time = ((i-1)*24*60)+8*60,
                          stop_time = ((i-1)*24*60)+18*60,
                          dist = function() c(0,rexp(10*10, rate = taxa_contacto),-1)),
                  mon = 2)

    gestao_contactos_SA %>% run(until = 5 * 24 * 60)  %>% invisible
})

# Métricas para a análise de eficiência
df <- metrics4.fun(gestao_contactos_SA_50)
ftable_2 <- ftable(df)

ftable_2 <- ftable(ftable_2, bg = "#0D2B6", part = "header")
ftable_2 <- color(ftable_2, color = "white", part = "header")
ftable_2 <- bold(ftable_2, bold = TRUE, part="header")
ftable_2 <- autofit(ftable_2)
ftable_2 %>%
  bold(part = "body", j= 1, j = 1) %>%
  hline(i = 1, part = "body", border = fp_border(color = "#002060", style = "dashed", width = 1)) %>%
  bold(part = "body", j= 4, j = 4) %>%
  hline(i = 4, part = "body", border = fp_border(color = "#002060", style = "dashed", width = 1)) %>%
  bold(part = "body", j= 5, j = 5) %>%
  hline(i = 5, part = "body", border = fp_border(color = "#002060", style = "dashed", width = 1)) %>%
  bold(part = "body", j= 11, j = 11) %>%
  hline(i = 12, part = "body", border = fp_border(color = "#002060", style = "solid", width = 1))
```

Métrica	Valor	%
Total de Pacientes	1971	
Total de Pacientes Atendidos	1021	51.8
Pacientes Totais Não Atendidos	950	48.2
Total de Chamadas	1672	
Chamadas Telefónicas Atendidas	722	43.2
Chamadas Perdidas	950	56.8
Total de Mensagens	299	
Mensagens Respondidas	298	99.9
Mensagens Não Respondidas	0	0.1
Pacientes Consulta Presencial	101	9.9
Pacientes Consulta Telefónica	613	60.0
Pacientes Sem Necessidade	307	30.1
Tempo Médio dos Pacientes Atendidos (minutos)	**Média**	**DP**
Tempo de Espera Médio	4.82	0.5
Tempo de Espera Médio [Chamadas]	4.21	0.0
Tempo de Espera Médio [Mensagens]	8.25	3.1
Tempo de Serviço Médio	5.70	0.0
Tempo Total de Processo	7.78	0.5

Tabela 2 - Variáveis de análise da eficiência da 2ª Simulação - **Situação Atual** I 50 Réplicas.

Replicando o mesmo cenário 50 vezes, adquirimos uma visão mais robusta e ampla do processo de gestão de contactos pré-operatórios, contudo os resultados das métricas, sendo a média das 50 réplicas, pouco oscilaram face aos anteriores. (**Tabela 2**)

```
gestao_contactos_SA_50 %>% get_mon_resources() %>% plot(metric = "utilization")
gestao_contactos_SA_50 %>% get_mon_resources() %>% plot(metric = "usage", steps=T)

# De forma a tratar primeiro a 'waiting_time' antes de a visualizar graficamente, optamos por fazer o gráfico de raiz, utili
zando o código da biblioteca no mesmo
# Fonte: https://github.com/r-simmer/simmer.plot/blob/7244b92cf0e0ba05a8d22409b1239f8972335/R/plot.arrivals.R
gestao_contactos_SA_50_filt <- gestao_contactos_SA_50 %>%
  get_mon_arrivals() %>%
  dplyr::mutate(waiting_time = end_time - start_time - activity_time) %>%
  transform(waiting_time = ifelse(waiting_time > 840, waiting_time - 840, waiting_time))

ggplot(gestao_contactos_SA_50_filt) +
  ecdf(x = gestao_contactos_SA_50_filt$end_time, y = gestao_contactos_SA_50_filt$waiting_time) +
  geom_line(aes(group = gestao_contactos_SA_50_filt$replication), alpha = 1.0 / (log(max(gestao_contactos_SA_50_filt$repli
cation)) + 1)) +
  stat_smooth() +
  xlab("simulation time") +
  ylab("waiting time") +
  ggtitle("Waiting time evolution") +
  expand_limits(y = 0)

gestao_contactos_SA_50 %>% get_mon_arrivals() %>% plot(metric = "activity_time")
```

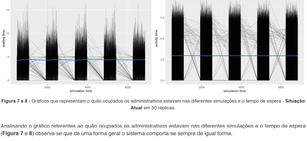


Figura 7 e 8 - Gráficos que representam o quão ocupados os administrativos estavam nas diferentes simulações e o tempo de espera - **Situação Atual** em 50 réplicas.

Analisando o gráfico referentes ao quão ocupados os administrativos estavam nas diferentes simulações e o tempo de espera (**Figura 7** e **8**) observa-se que de uma forma geral o sistema comporta-se sempre de igual forma.

Sendo que cerca de 50% das chamadas efetuadas pelos pacientes não são respondidas atempadamente (dentro dos 5 minutos aceitáveis), foram apliczadas possibilidades de melhoria do processo.

Situação de Melhoria

Possibilidades de Melhoria do Processo de Gestão de Contactos

As melhorias aplicadas em ambos cenários que seguem, envolvem a definição das distribuições do tempo entre contactos estando apresentadas na **Tabela 3** e alteração de mais administrativos em diferentes períodos evidenciadas na **Tabela 4**.

Tabela 3 - Número de contactos por período.

Período	Número de Contactos
8h – 9h	100
9h – 9h30	2
9h30 – 12h	4
12h – 19h30	3
19h30 – 18h	1

Tabela 4 - Número de contactos por período.

Período	Número de Administrativos
8h – 9h	1
9h – 9h30	2
9h30 – 12h	4
12h – 19h30	3
19h30 – 18h	1

Situação de Melhoria (Cenário `a`) | 1 Réplica

Ademais, no **cenário A** foi acrescida uma particularidade: a prioridade das chamadas sobre as mensagens sendo que um administrativo poderá interromper a análise de uma mensagem para atender uma chamada. Para tal, quando é decidido no sistema o tipo do contacto do paciente, foi acrescentado na trajetória do contacto por chamada a função `set_priorization` com os parâmetros: `priority=1`, `preemptive=2`, `restart=FALSE`, e foi acrescentado no `add_resource` o `preemptive = TRUE`.

```
##############################  Exercício 2  ##############################
#                              Cenário a)  ------------------------------
set.seed(2023)

# Trajetória para escolher o tipo de contacto com base nas probabilidades, considerando a prioridade das chamadas e a sua ordem
# ----------------------------------------------------------------------
tipo_contacto_prioritario_a <-   trajectory("Tipo de Contacto") %>%
  branch(function() {rnorm(1) > 0.80} * 1, continue = c(FALSE, FALSE),

         # Trajetória caso o contacto tenha sido por telefone
         trajectory() %>%
         # values = c(priority, preemptible, restart)
         # fonte: https://r-simmer.org/reference/set_priorization
         set_prioritization(function() {c(priority=1, preemptive=2, restart=FALSE)}) %>%
         join(contacto_pacientes_tel),

         # Trajetória caso o contacto tenha sido por via eletrónica
         contacto_pacientes_ems)


# Definir o agendamento dos administrativos
administrativos_schedule <- schedule(timetable = c(7*60, 9*60, 9.5*60, 11*60, 15.5*60, 18*60),
                                     values = c(1, 2, 4, 3, 1, 0),
                                     period = 24*60)

# Criar um ambiente de simulação, replicando o sistema apenas 1 vez
gestao_contactos_SM_1a <- simmer("Gestao dos Contactos | ReSituação 1 a)") %>%

  add_resource("administration", administrativos_schedule, preemptive = TRUE)

  # Percurso dos dia 1a 1dh
  for(j IN 1:5) gestao_contactos_SM_1a %>%
    add_generator(paste("Patiente_M",j,sep=""), trajectory = tipo_contacto_prioritario_a,
                  from_to(start_time = (((M-1)*24*60)+(8*60),
                          stop_time = ((M-1)*24*60)+(9*60),
                          dist = function() {c(0, rexp(100, rate = 60/60), -1)},
                          ),
                  mon = 2)

  # Percurso dos 1dh 1a 2dh
  for(j IN 1:15) gestao_contactos_SM_1a %>%
    add_generator(paste("Patiente_T",j,sep=""), trajectory = tipo_contacto_prioritario_a,
                  from_to(start_time = (((M-1)*24*60)+(9*60),
                          stop_time = ((M-1)*24*60)+(9*60+30),
                          dist = function() {c(0, rexp(360, rate = 60/60), -1)},
                          ),
                  mon = 2)

  # Percurso dos 2dh 1a 3dh
  for(j IN 1:40) gestao_contactos_SM_1a %>%
    add_generator(paste("Patiente_N",j,sep=""), trajectory = tipo_contacto_prioritario_a,
                  from_to(start_time = (((M-1)*24*60)+(9*60+30),
                          stop_time = ((M-1)*24*60)+(12*60),
                          dist = function() {c(0, rexp(120, rate = 60/60), -1)},
                          ),
                  mon = 2)

  # Percurso dos 3dh 1a 4dh
  for(j IN 1:12) gestao_contactos_SM_1a %>%
    add_generator(paste("Patiente_O",j,sep=""), trajectory = tipo_contacto_prioritario_a,
                  from_to(start_time = (((M-1)*24*60)+(12*60),
                          stop_time = (((M-1)*24*60)+(19*60),
                          dist = function() {c(0, rexp(360, rate = 60/60), -1)},
                          ),
                  mon = 2)

  # Percurso dos 4dh 1a 5dh
  for(j IN 1:3) gestao_contactos_SM_1a %>%
    add_generator(paste("Patiente_P",j,sep=""), trajectory = tipo_contacto_prioritario_a,
                  from_to(start_time = (((M-1)*24*60)+(19*60),
                          stop_time = (((M-1)*24*60)+(19*60+30),
                          dist = function() {c(0, rexp(60, rate = 20/60), -1)},
                          ),
                  mon = 2)

  gestao_contactos_SM_1a %>% run(until = 5*24*60) %>% invisible

# Métricas para a análise da eficiência (Exercício 2 | Cenário a)
df <- as.data.frame(metrics_fun(gestao_contactos_SM_1a))
ftable_a <- ftable(df)

ftable_a <- bg(ftable_a, bg = "#0B2D60", part = "header")
ftable_a <- color(ftable_a, color = "white", part = "header")
ftable_a <- autofit(ftable_a, bold = TRUE, part="header")
ftable_a <- autofit(ftable_a)
ftable_a %>%
  bold(part = "body", j= 1, j = 1) %>%
  hline(i = 1, part = "body", border = fp_border(color = "#0B2D60", style = "dashed", width = 1)) %>%
  bold(part = "body", i= 4, j = 1) %>%
  hline(i = 4, part = "body", border = fp_border(color = "#0B2D60", style = "dashed", width = 1)) %>%
  bold(part = "body", i= 7, j = 1) %>%
  hline(i = 7, part = "body", border = fp_border(color = "#0B2D60", style = "dashed", width = 1)) %>%
  bold(part = "body", i= 13, j = 1) %>%
  hline(i = 13, part = "body", border = fp_border(color = "#0B2D60", style = "solid", width = 1))
```

Métrica	Valor	%
Total de Pacientes	1976	
Total de Pacientes Atendidos	1318	66.7
Pacientes Totais Não Atendidos	658	33.3
Total de Chamadas	1704	
Chamadas Telefónicas Atendidas	1135	66.6
Chamadas Perdidas	569	33.4
Total de Mensagens	272	
Mensagens Respondidas	183	67.3
Mensagens Não Respondidas	89	32.7
Pacientes Consulta Presencial	133	10.1
Pacientes Consulta Telefónica	762	57.8
Pacientes Sem Necessidade	423	32.1
Tempo Médio dos Pacientes Atendidos (minutos)	**Média**	**DP**
Tempo de Espera Médio	112.20	401.2
Tempo de Espera Médio [Chamadas]	4.10	28.9
Tempo de Espera Médio [Mensagens]	1117.72	730.5
Tempo de Serviço Médio	5.69	1.2
Tempo Total de Processo	116.17	401.7

Tabela 5 - Variáveis de análise da eficiência da 3ª Simulação - **Situação de Melhoria** - Cenário `a)` | 1 Réplica.

Desta forma, obteve-se algumas melhorias no que toca às métricas de análise de eficiência do sistema, passando de 48% para 33% de pacientes não atendidos. Em contraparte, em média, o tempo de espera sofreu um grande aumento, sendo agora aproximadamente 58 min., já o tempo de serviço manteve-se. Além disso, as mensagens não são respondidas na sua totalidade tendo cerca de 33% por responder, o que não é significativo sendo que as mensagens apenas representam 15% dos contactos. (**Tabela 5**)

```
gestao_contactos_SM_1a %>% get_mon_resources() %>% plot(metric = "utilization")
gestao_contactos_SM_1a %>% get_mon_resources() %>% plot(metric = "usage", steps=T)
```

Figura 9 e 10 - Gráficos que representa o quanto os recursos são utilizado (esquerda); quando os recursos (administrativos) estão ocupados, bem como quantas pacientes estão na fila a cada momento (direita) na **Situação de Melhoria - Cenário** `a)` sem 1 réplica.

Ao analisar a **Figura 9** e **10**, pode-se averiguar que os administrativos estão a trabalhar durante todo o tempo e, ao verificar o funcionamento do sistema, este reflete uma grande quantia de mensagens por responder de um dia para o outro.

Situação de Melhoria (Cenário `a`) | 50 Réplicas

```
# Criar um ambiente de simulação, replicando o sistema 50 vezes das Possibilidades de Melhoria | Cenário a) em 50 réplicas
set.seed(2023)

gestao_contactos_SM_50a <-
  lapply(1:50, function(i){
    gestao_contactos_SM_a <- simmer(paste("Gestao dos Contactos a) 50rep", i)) %>%
    add_resource("administration", administrativos_schedule, preemptive = TRUE)

  # Percurso dos dia 1a 1dh
  for(j IN 1:5) gestao_contactos_SM_a %>%
    add_generator(paste("Patiente_M",j,sep=""), trajectory = tipo_contacto_prioritario_a,
                  from_to(start_time = (((M-1)*24*60)+(8*60),
                          stop_time = ((M-1)*24*60)+(9*60),
                          dist = function() {c(0, rexp(100, rate = 60/60), -1)},
                          ),
                  mon = 2)

  # Percurso dos 1dh 1a 2dh
  for(j IN 1:15) gestao_contactos_SM_a %>%
    add_generator(paste("Patiente_T",j,sep=""), trajectory = tipo_contacto_prioritario_a,
                  from_to(start_time = (((M-1)*24*60)+(9*60),
                          stop_time = ((M-1)*24*60)+(9*60+30),
                          dist = function() {c(0, rexp(360, rate = 60/60), -1)},
                          ),
                  mon = 2)

  # Percurso dos 2dh 1a 3dh
  for(j IN 1:40) gestao_contactos_SM_a %>%
    add_generator(paste("Patiente_N",j,sep=""), trajectory = tipo_contacto_prioritario_a,
                  from_to(start_time = (((M-1)*24*60)+(9*60+30),
                          stop_time = ((M-1)*24*60)+(12*60),
                          dist = function() {c(0, rexp(120, rate = 60/60), -1)},
                          ),
                  mon = 2)

  # Percurso dos 3dh 1a 4dh
  for(j IN 1:12) gestao_contactos_SM_a %>%
    add_generator(paste("Patiente_O",j,sep=""), trajectory = tipo_contacto_prioritario_a,
                  from_to(start_time = (((M-1)*24*60)+(12*60),
                          stop_time = (((M-1)*24*60)+(19*60),
                          dist = function() {c(0, rexp(360, rate = 60/60), -1)},
                          ),
                  mon = 2)

  # Percurso dos 4dh 1a 5dh
  for(j IN 1:3) gestao_contactos_SM_a %>%
    add_generator(paste("Patiente_P",j,sep=""), trajectory = tipo_contacto_prioritario_a,
                  from_to(start_time = (((M-1)*24*60)+(19*60),
                          stop_time = (((M-1)*24*60)+(19*60+30),
                          dist = function() {c(0, rexp(60, rate = 20/60), -1)},
                          ),
                  mon = 2)

  gestao_contactos_SM_a %>% run(until = 5*24*60) %>% invisible
})

# Métricas para a análise da eficiência (Exercício 2 | Cenário b) 50 réplicas)
df <- as.data.frame(metrics_fun(gestao_contactos_SM_50a))
ftable_a <- ftable(df)

ftable_a <- bg(ftable_a, bg = "#0B2D60", part = "header")
ftable_a <- color(ftable_a, color = "white", part = "header")
ftable_a <- bold(ftable_a, bold = TRUE, part="header")
ftable_a <- autofit(ftable_a)
ftable_a %>%
  bold(part = "body", j= 1, j = 1) %>%
  hline(i = 1, part = "body", border = fp_border(color = "#0B2D60", style = "dashed", width = 1)) %>%
  bold(part = "body", i= 4, j = 1) %>%
  hline(i = 4, part = "body", border = fp_border(color = "#0B2D60", style = "dashed", width = 1)) %>%
  bold(part = "body", i= 7, j = 1) %>%
  hline(i = 7, part = "body", border = fp_border(color = "#0B2D60", style = "dashed", width = 1)) %>%
  bold(part = "body", i= 13, j = 1) %>%
  hline(i = 13, part = "body", border = fp_border(color = "#0B2D60", style = "solid", width = 1))
```

Métrica	Valor	%
Total de Pacientes	1971	
Total de Pacientes Atendidos	1313	66.6
Pacientes Totais Não Atendidos	658	33.4
Total de Chamadas	1676	
Chamadas Telefónicas Atendidas	1111	66.3
Chamadas Perdidas	566	33.7
Total de Mensagens	295	
Mensagens Respondidas	203	68.7
Mensagens Não Respondidas	92	31.3
Pacientes Consulta Presencial	131	10.0
Pacientes Consulta Telefónica	787	59.9
Pacientes Sem Necessidade	395	30.1
Tempo Médio dos Pacientes Atendidos (minutos)	**Média**	**DP**
Tempo de Espera Médio	128.72	27.5
Tempo de Espera Médio [Chamadas]	4.21	0.5
Tempo de Espera Médio [Mensagens]	1171.53	294.9
Tempo de Serviço Médio	5.70	0.0
Tempo de Processo	132.71	27.5

Tabela 6 - Variáveis de análise da eficiência da 4ª Simulação - **Situação de Melhoria** - Cenário `a)` | 50 Réplicas.

Uma vez mais, ao replicar o mesmo cenário 50 vezes, os resultados das métricas, não apresentaram oscilações notáveis face aos anteriores, à exceção do tempo do ele espera médio que aumenta cerca de 10 minutos. (Tabela 6)

Em relação à ocupação dos administrativos, ao comparar com as 50 réplicas da situação atual, é notável a diminuição da mesma. Já sobre o tempo de espera de um para outro não é comparável sendo que a mensuração do mesmo foi feita de forma diferente, uma vez que neste caso, no **cenário A**, definiu-se um número de administrativos para cada intervalo do tempo.

```
gestao_contactos_SM_50a %>% get_mon_resources() %>% plot(metric = "utilization")
gestao_contactos_SM_50a %>% get_mon_resources() %>% plot(metric = "usage", steps=T)
gestao_contactos_SM_50a %>% get_mon_arrivals() %>% plot(metric = "waiting_time")
gestao_contactos_SM_50a %>% get_mon_arrivals() %>% plot(metric = "activity_time")
```

Figura 11 e 12 - Gráficos que representa o quanto os recursos são utilizado (esquerda); quando os recursos (administrativos) estão ocupados, bem como quantas pacientes estão na fila a cada momento (direita) na **Situação de Melhoria - Cenário** `a)` em 50 réplicas.

Figura 13 e 14 - Gráficos que representam quanto tempo os pacientes estão na fila e o tempo do seu atendimento na **Situação de Melhoria - Cenário** `a)` em 50 réplicas.

Situação de Melhoria (Cenário `b`) | 1 Réplica

O **cenário B** obteve do **cenário A** na medida em que os administrativos atendem as chamadas telefónicas sem interromper a análise das mensagens. Para isso, substitui-se o `set_priorization` com o parâmetro `preemptive=1`, e acrescentou-se no `add_resource` o `preemptive = FALSE`, para que seja dada prioridade à mensagem.

```
#                              Cenário b)  ------------------------------
set.seed(2023)

# Trajetória para escolher o tipo de contacto com base nas probabilidades, considerando apenas a prioridade das chamadas
tipo_contacto_prioritario_b <-   trajectory("Tipo de Contacto") %>%
  branch(function() {rnorm(1) > 0.80} * 1, continue = c(FALSE, FALSE),

         # Trajetória caso o contacto tenha sido por telefone
         trajectory() %>%
         # values = c(priority, preemptible, restart)
         set_prioritization(function() {c(priority=1, preemptive=1, restart=FALSE)}) %>%
         join(contacto_pacientes_tel),

         # Trajetória caso o contacto tenha sido por via eletrónica
         contacto_pacientes_ems)
```

Situação de Melhoria (Cenário `b`) | 50 Réplicas

```
# Criar um ambiente de simulação, replicando o sistema 50 vezes das Possibilidades de Melhoria | Cenário b) em 50 réplicas
set.seed(2023)

gestao_contactos_SM_50b <-
  lapply(1:50, function(i){
    gestao_contactos_SM_b <- simmer(paste("Gestao dos Contactos b) 50rep", i)) %>%
    add_resource("administration", administrativos_schedule, preemptive = TRUE)

  # Percurso dos dia 1a 1dh
  for(j IN 1:5) gestao_contactos_SM_b %>%
    add_generator(paste("Patiente_M",j,sep=""), trajectory = tipo_contacto_prioritario_b,
                  from_to(start_time = (((M-1)*24*60)+(8*60),
                          stop_time = ((M-1)*24*60)+(9*60),
                          dist = function() {c(0, rexp(100, rate = 60/60), -1)},
                          ),
                  mon = 2)

  # Percurso dos 1dh 1a 2dh
  for(j IN 1:15) gestao_contactos_SM_b %>%
    add_generator(paste("Patiente_T",j,sep=""), trajectory = tipo_contacto_prioritario_b,
                  from_to(start_time = (((M-1)*24*60)+(9*60),
                          stop_time = ((M-1)*24*60)+(9*60+30),
                          dist = function() {c(0, rexp(360, rate = 60/60), -1)},
                          ),
                  mon = 2)

  # Percurso dos 2dh 1a 3dh
  for(j IN 1:40) gestao_contactos_SM_b %>%
    add_generator(paste("Patiente_N",j,sep=""), trajectory = tipo_contacto_prioritario_b,
                  from_to(start_time = (((M-1)*24*60)+(9*60+30),
                          stop_time = ((M-1)*24*60)+(12*60),
                          dist = function() {c(0, rexp(120, rate = 60/60), -1)},
                          ),
                  mon = 2)

  # Percurso dos 3dh 1a 4dh
  for(j IN 1:12) gestao_contactos_SM_b %>%
    add_generator(paste("Patiente_O",j,sep=""), trajectory = tipo_contacto_prioritario_b,
                  from_to(start_time = (((M-1)*24*60)+(12*60),
                          stop_time = (((M-1)*24*60)+(19*60),
                          dist = function() {c(0, rexp(360, rate = 60/60), -1)},
                          ),
                  mon = 2)

  # Percurso dos 4dh 1a 5dh
  for(j IN 1:3) gestao_contactos_SM_b %>%
    add_generator(paste("Patiente_P",j,sep=""), trajectory = tipo_contacto_prioritario_b,
                  from_to(start_time = (((M-1)*24*60)+(19*60),
                          stop_time = (((M-1)*24*60)+(19*60+30),
                          dist = function() {c(0, rexp(60, rate = 20/60), -1)},
                          ),
                  mon = 2)

  gestao_contactos_SM_b %>% run(until = 5*24*60) %>% invisible
})

# Métricas para a análise da eficiência (Exercício 2 | Cenário b) 50 réplicas)
df <- as.data.frame(metrics_fun(gestao_contactos_SM_50b))
ftable_a <- ftable(df)

ftable_a <- bg(ftable_a, bg = "#0B2D60", part = "header")
ftable_a <- color(ftable_a, color = "white", part = "header")
ftable_a <- bold(ftable_a, bold = TRUE, part="header")
ftable_a <- autofit(ftable_a)
ftable_a %>%
  bold(part = "body", j= 1, j = 1) %>%
  hline(i = 1, part = "body", border = fp_border(color = "#0B2D60", style = "dashed", width = 1)) %>%
  bold(part = "body", i= 4, j = 1) %>%
  hline(i = 4, part = "body", border = fp_border(color = "#0B2D60", style = "dashed", width = 1)) %>%
  bold(part = "body", i= 7, j = 1) %>%
  hline(i = 7, part = "body", border = fp_border(color = "#0B2D60", style = "dashed", width = 1)) %>%
  bold(part = "body", i= 13, j = 1) %>%
  hline(i = 13, part = "body", border = fp_border(color = "#0B2D60", style = "solid", width = 1))
```

Métrica	Valor	%
Total de Pacientes	1971	
Total de Pacientes Atendidos	1313	66.6
Pacientes Totais Não Atendidos	658	33.4
Total de Chamadas	1676	
Chamadas Telefónicas Atendidas	1111	66.3
Chamadas Perdidas	566	33.7
Total de Mensagens	295	
Mensagens Respondidas	203	68.7
Mensagens Não Respondidas	92	31.3
Pacientes Consulta Presencial	131	10.0
Pacientes Consulta Telefónica	787	59.9
Pacientes Sem Necessidade	395	30.1
Tempo Médio dos Pacientes Atendidos (minutos)	**Média**	**DP**
Tempo de Espera Médio	128.72	27.5
Tempo de Espera Médio [Chamadas]	4.21	0.5
Tempo de Espera Médio [Mensagens]	1171.53	294.9
Tempo de Serviço Médio	5.70	0.0
Tempo de Processo	132.71	27.5

```
        set_priorities(priorities) c(priority=1, preemptive=1, restart=FALSE) %>%
        join(contacts_patients,T4),

        # Trajetória caso o contacto tenha sido por via eletrónica
        contacts_patientes_em)

# Definir o agendamento dos administrativos
administrativos_schedule <- schedule(timetable = c(7*60, 9*60, 9.5*60, 12*60, 15.5*60, 18*60),
                                     values = c(1, 2, 4, 2, 1, 0),
                                     period = 24*60)

# Criar um ambiente de simulação, replicando o sistema na Situação de Melhoria (SM) apenas 1 vez
gestao_contactos_SM_1b <- simmer("Gestao dos Contactos | Melhoria 1 b)") %>%

  add_resource("administration", administrativos_schedule, preemptive = FALSE)

  # Período das 8h às 18h
  for(j0 in 1:5) gestao_contactos_SM_1b %>%
    add_generator(paste("Paciente_M",j0), trajectory = tipo_contacto_prioritario_b,
                  from_to(start.time = ((j0-1)*1*24*60+8*60,
                          stop.time = ((j0-1)*1*24*60+18*60,
                          dist = function() {c(0, rexp(120, rate = 60/60), -1)},
                          ),
                  mon = 2 )

  # Período das 18h às 1dh
  for(j7 in 1:5) gestao_contactos_SM_1b %>%
    add_generator(paste("Paciente_T",j7), trajectory = tipo_contacto_prioritario_b,
                  from_to(start.time = ((j7-1)*1*24*60+18*60,
                          stop.time = ((j7-1)*1*24*60+1d*60,
                          dist = function() {c(0, rexp(360, rate = 60/60), -1)},
                          ),
                  mon = 2 )

  # Período das 1dh às 1dh
  for(j8 in 1:5) gestao_contactos_SM_1b %>%
    add_generator(paste("Paciente_N",j8), trajectory = tipo_contacto_prioritario_b,
                  from_to(start.time = ((j8-1)*1*24*60+1d*60,
                          stop.time = ((j8-1)*1*24*60+8*60,
                          dist = function() {c(0, rexp(480, rate = 20/60), -1)},
                          ),
                  mon = 2 )

gestao_contactos_SM_1b %>% run(until = 5*24*60) %>% invisible

# Métricas para a análise da eficiência (Exercício 2 (Cenário b) )
df <- as.data.frame(metrics_fun(gestao_contactos_SM_1b))
ftable_b <- ftable(df)

ftable_b <- bg(ftable_b, bg = "#0B2060", part = "header")
ftable_b <- color(ftable_b, color = "white", part = "header")
ftable_b <- bold(ftable_b, bold = TRUE, part="header")
ftable_b <- autofit(ftable_b)
ftable_b %>%
  hline(part = "body", i= 1, j = 1) %>%
  hline(i = 1, part = "body", border = fp_border(color = "#0B2060", style = "dashed", width = 1)) %>%
  hline(part = "body", i= 4, j = 1) %>%
  hline(i = 4, part = "body", border = fp_border(color = "#0B2060", style = "dashed", width = 1)) %>%
  hline(part = "body", i= 7, j = 1) %>%
  hline(i = 7, part = "body", border = fp_border(color = "#0B2060", style = "dashed", width = 1)) %>%
  hline(part = "body", i= 11, j = 1) %>%
  hline(i = 11, part = "body", border = fp_border(color = "#0B2060", style = "solid", width = 1))
```

Métrica	Valor	%
Total de Pacientes	1977	
Total de Pacientes Atendidos	1329	67.2
Pacientes Totais Não Atendidos	648	32.8
Total de Chamadas	1699	
Chamadas Telefónicas Atendidas	1086	63.9
Chamadas Perdidas	613	36.1
Total de Mensagens	278	
Mensagens Respondidas	243	87.4
Mensagens Não Respondidas	35	12.6
Pacientes Consulta Presencial	147	11.1
Pacientes Consulta Telefónica	742	55.8
Pacientes Sem Necessidade	440	33.1
Tempo Médio dos Pacientes Atendidos (minutos)	**Média**	**DP**
Tempo de Espera Médio	69.47	243.4
Tempo de Espera Médio [Chamadas]	3.60	1.5
Tempo de Espera Médio [Mensagens]	530.01	481.3
Tempo de Serviço Médio	5.65	1.2
Tempo Total de Processo	73.34	243.9

Tabela 7 - Variáveis de análise da eficiência da 5ª Simulação na **Situação de Melhoria - Cenário b)** em 1 réplica.

Relativamente aos resultados obtidos, os pacientes não atendidos diminuiram ligeiramente, tendo nos 32.8%. Por sua vez, face ao **cenário A**, o número de chamadas telefónicas perdidas aumentou ligeiramente tendo em 36.1%, mas o número de mensagens não respondidas diminui substancialmente ficando nos 12.6%. O tempo de espera médio, também reduziu passando a ser de 32.7 min, o tempo de serviço médio manteve-se constante (**Tabela 7**).

```
gestao_contactos_SM_1b %>% get_mon_resources() %>% plot(metric = "utilization")
gestao_contactos_SM_1b %>% get_mon_resources() %>% plot(metric = "usage", steps=T)
```

Figura 15 e 16 - Gráficos que representa o quanto os recursos são utilizado (esquerda); quando os recursos (administrativos) estão ocupados, bem como quantas pacientes estão na fila a cada momento (direita) na **Situação de Melhoria - Cenário b)** em 1 Réplica.

Os administrativos são utilizados a 100%, pelo que podem representar estrangulamentos no sistema, ou seja, se nas mesmas condições tivessemos mais administrativos poderiamos melhorar os resultados obtidos (**Figura 13**). Na **Figura 14**, vemos que os administrativos estão sempre a ser utilizados e que o número de pacientes à espera de serem atendidos tende a aumentar ao longo do tempo.

Situação de Melhoria (Cenário `b)`) I 50 Réplicas

```
# Criar um ambiente de simulação, replicando o sistema 50 vezes das Possibilidades de Melhoria (Cenário b) em 50 réplicas
set.seed(2023)

gestao_contactos_SM_50b <-
  lapply(1:50, function(i){
    gestao_contactos_SM_b <- simmer(paste("Gestao dos Contactos SMmp", i)) %>%
    add_resource("administration", administrativos_schedule, preemptive = FALSE)

    # Período das 8h às 18h
    for(j0 in 1:5) gestao_contactos_SM_b %>%
      add_generator(paste("Paciente_M",j0), trajectory = tipo_contacto_prioritario_b,
                    from_to(start.time = ((j0-1)*1*24*60+8*60,
                            stop.time = ((j0-1)*1*24*60+18*60,
                            dist = function() {c(0, rexp(120, rate = 60/60), -1)},
                            ),
                    mon = 2)

    # Período das 18h às 1dh
    for(j7 in 1:5) gestao_contactos_SM_b %>%
      add_generator(paste("Paciente_T",j7), trajectory = tipo_contacto_prioritario_b,
                    from_to(start.time = ((j7-1)*1*24*60+18*60,
                            stop.time = ((j7-1)*1*24*60+1d*60,
                            dist = function() {c(0, rexp(360, rate = 60/60), -1)},
                            ),
                    mon = 2)

    # Período das 1dh às 1dh
    for(j8 in 1:5) gestao_contactos_SM_b %>%
      add_generator(paste("Paciente_N",j8), trajectory = tipo_contacto_prioritario_b,
                    from_to(start.time = ((j8-1)*1*24*60+1d*60,
                            stop.time = ((j8-1)*1*24*60+8*60,
                            dist = function() {c(0, rexp(480, rate = 20/60), -1)},
                            ),
                    mon = 2)

    gestao_contactos_SM_b %>% run(until = 5*24*60) %>% invisible
})

# Métricas para a análise da eficiência (Exercício 2 (Cenário b) 50 réplicas)
df <- as.data.frame(metrics_fun(gestao_contactos_SM_50b))
ftable_b <- ftable(df)

ftable_b <- bg(ftable_b, bg = "#0B2060", part = "header")
ftable_b <- color(ftable_b, color = "white", part = "header")
ftable_b <- bold(ftable_b, bold = TRUE, part="header")
ftable_b <- autofit(ftable_b)
ftable_b %>%
  hline(part = "body", i= 1, j = 1) %>%
  hline(i = 1, part = "body", border = fp_border(color = "#0B2060", style = "dashed", width = 1)) %>%
  hline(part = "body", i= 4, j = 1) %>%
  hline(i = 4, part = "body", border = fp_border(color = "#0B2060", style = "dashed", width = 1)) %>%
  hline(part = "body", i= 7, j = 1) %>%
  hline(i = 7, part = "body", border = fp_border(color = "#0B2060", style = "dashed", width = 1)) %>%
  hline(part = "body", i= 11, j = 1) %>%
  hline(i = 11, part = "body", border = fp_border(color = "#0B2060", style = "solid", width = 1))
```

Métrica	Valor	%
Total de Pacientes	1969	
Total de Pacientes Atendidos	1319	67.0
Pacientes Totais Não Atendidos	650	33.0
Total de Chamadas	1677	
Chamadas Telefónicas Atendidas	1070	63.8
Chamadas Perdidas	607	36.2
Total de Mensagens	292	
Mensagens Respondidas	249	85.4
Mensagens Não Respondidas	43	14.6
Pacientes Consulta Presencial	135	10.2
Pacientes Consulta Telefónica	787	59.7
Pacientes Sem Necessidade	397	30.1
Tempo Médio dos Pacientes Atendidos (minutos)	**Média**	**DP**
Tempo de Espera Médio	79.05	26.9
Tempo de Espera Médio [Chamadas]	3.59	0.0
Tempo de Espera Médio [Mensagens]	592.53	224.5
Tempo de Serviço Médio	5.70	0.0
Tempo Total de Processo	82.96	26.9

Tabela 8 - Variáveis de análise da eficiência da 6ª Simulação - **Situação de Melhoria - Cenário b)** I 50 Réplicas.

Ao replicarmos o **cenário B** 50 vezes, os resultados obtidos mantiveram-se sensivelmente os mesmos, exceto o número de mensagens não respondidas que aumentou cerca de 2% e o tempo de espera médio que aumentou em 6 min. (**Tabela 8**).

```
gestao_contactos_SM_50b %>% get_mon_resources() %>% plot(metric = "utilization")
gestao_contactos_SM_50b %>% get_mon_resources() %>% plot(metric = "usage", steps=T)
gestao_contactos_SM_50b %>% get_mon_arrivals() %>% plot(metric = "waiting_time")
gestao_contactos_SM_50b %>% get_mon_arrivals() %>% plot(metric = "activity_time")
```

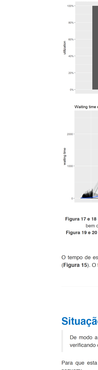


Figura 17 e 18 - Gráficos que representa o quanto os recursos são utilizado (esquerda); quando os recursos (administrativos) estão ocupados, bem como quantas pacientes estão na fila a cada momento (direita) na **Situação de Melhoria - Cenário b)** em 1 Réplica.
Figura 19 e 20 - Gráficos que representam quanto tempo os pacientes estão na fila e o tempo do seu atendimento na **Situação de Melhoria - Cenário b)** em 50 réplicas.

O tempo de espera tem tendência crescente, sendo que no início do ultimo dia existe um aumento substancial do seu valor (**Figura 15**). O tempo de atendimento tende a não sofrer alterações ao longo da simulação (**Figura 16**).

Situação de Melhoria [`EXTRA`]

De modo a tentar melhor o panorama geral da gestão de contactos, testámos diferentes vetores nos administrativos, verificando o seu impacto e potencial contribuição na otimização da eficiência do funcionamento do sistema.

Para que esta alocação fosse possível, a diretora dos serviços administrativos poderia escolher uma das opções que se seguem:

1. Inserir um sistema de rotatividade de turnos de forma a poder ter mais administrativos por período de tempo;
2. Contratar mais administrativos para o contacto com os pacientes;
3. Alocar mais administrativos de outros serviços à gestão de contactos pré-operatórios do serviço de cirurgia.

Situação de Melhoria (Cenário `Extra 1)`) I 1 Réplica

```
# ------------------------- Cenário Extra 1 -------------------------
# Trajetória para escolher o tipo de contacto com base nos prioridades, considerando apenas a prioridade das chamadas
set.seed(2023)

# Definir o agendamento dos administrativos
administrativos_schedule_extra1 <- schedule(timetable = c(7*60, 9*60, 9.5*60, 12*60, 15.5*60, 18*60),
                                            values = c(1, 2, 4, 2, 2, 0),
                                            # período a responder as mensagens que ficaram por responder do dia anterior.
                                            period = 24*60)

# Criar um ambiente de simulação, replicando o sistema na Situação de Melhoria Extra 1 (SM_E1), apenas 1 vez
gestao_contactos_SM_E1 <-
  lapply(1:50, function(i){
    gestao_contactos_SM_E1 <- simmer(paste("Gestao dos Contactos | Melhoria Extra 1 )", i)) %>%
    add_resource("administration", administrativos_schedule_extra1, preemptive = TRUE)

    # Período das 8h às 18h
    for(j0 in 1:5) gestao_contactos_SM_E1 %>%
      add_generator(paste("Paciente_M",j0), trajectory = tipo_contacto_prioritario_a,
                    from_to(start.time = ((j0-1)*1*24*60+8*60,
                            stop.time = ((j0-1)*1*24*60+18*60,
                            dist = function() {c(0, rexp(120, rate = 60/60), -1)},
                            ),
                    mon = 2)

    # Período das 18h às 1dh
    for(j7 in 1:5) gestao_contactos_SM_E1 %>%
      add_generator(paste("Paciente_T",j7), trajectory = tipo_contacto_prioritario_a,
                    from_to(start.time = ((j7-1)*1*24*60+18*60,
                            stop.time = ((j7-1)*1*24*60+1d*60,
                            dist = function() {c(0, rexp(360, rate = 60/60), -1)},
                            ),
                    mon = 2)

    # Período das 1dh às 1dh
    for(j8 in 1:5) gestao_contactos_SM_E1 %>%
      add_generator(paste("Paciente_N",j8), trajectory = tipo_contacto_prioritario_a,
                    from_to(start.time = ((j8-1)*1*24*60+1d*60,
                            stop.time = ((j8-1)*1*24*60+8*60,
                            dist = function() {c(0, rexp(480, rate = 20/60), -1)},
                            ),
                    mon = 2)

    gestao_contactos_SM_E1 %>% run(until = 5*24*60) %>% invisible
})

# Métricas para a análise da eficiência (Exercício 2 (Cenário Extra 1) )
df <- as.data.frame(metrics_fun(gestao_contactos_SM_E1))
ftable_E1 <- ftable(df)

ftable_E1 <- bg(ftable_E1, bg = "#0B2060", part = "header")
ftable_E1 <- color(ftable_E1, color = "white", part = "header")
ftable_E1 <- bold(ftable_E1, bold = TRUE, part="header")
ftable_E1 <- autofit(ftable_E1)
ftable_E1 %>%
  hline(part = "body", i= 1, j = 1) %>%
  hline(i = 1, part = "body", border = fp_border(color = "#0B2060", style = "dashed", width = 1)) %>%
  hline(part = "body", i= 4, j = 1) %>%
  hline(i = 4, part = "body", border = fp_border(color = "#0B2060", style = "dashed", width = 1)) %>%
  hline(part = "body", i= 7, j = 1) %>%
  hline(i = 7, part = "body", border = fp_border(color = "#0B2060", style = "dashed", width = 1)) %>%
  hline(part = "body", i= 11, j = 1) %>%
  hline(i = 11, part = "body", border = fp_border(color = "#0B2060", style = "solid", width = 1))
```

Métrica	Valor	%
Total de Pacientes	1995	
Total de Pacientes Atendidos	1365	69.5
Pacientes Totais Não Atendidos	600	30.5
Total de Chamadas	1673	
Chamadas Telefónicas Atendidas	1148	68.6
Chamadas Perdidas	525	31.4
Total de Mensagens	293	
Mensagens Respondidas	217	74.1
Mensagens Não Respondidas	76	25.9
Pacientes Consulta Presencial	135	9.9
Pacientes Consulta Telefónica	823	60.3
Pacientes Sem Necessidade	407	29.8
Tempo Médio dos Pacientes Atendidos (minutos)	**Média**	**DP**
Tempo de Espera Médio	117.52	23.7

```
gestao_contactos_SM_E1 %>% get_mon_resources() %>% plot(metric = "utilization")
gestao_contactos_SM_E1 %>% get_mon_resources() %>% plot(metric = "usage", steps=T)
```

Tempo de Espera Médio [Chamadas]	4.82	0.8
Tempo de Espera Médio [Mensagens]	991.80	225.1
Tempo de Serviço Médio	5.70	0.0
Tempo Total De Processo	121.65	23.7

Tabela 9 - Variáveis de análise da eficiência da 3ª Simulação - **Situação de Melhoria - Cenário** `Extra 1` | 1 Réplica.

```
gestao_contactos_SM_50_E1 %>% get_mon_resources() %>% plot(metric = "utilization")
gestao_contactos_SM_50_E1 %>% get_mon_resources() %>% plot(metric = "usage", steps=T)
gestao_contactos_SM_50_E1 %>% get_mon_arrivals() %>% plot(metric = "waiting_time")
gestao_contactos_SM_50_E1 %>% get_mon_arrivals() %>% plot(metric = "activity_time")
```

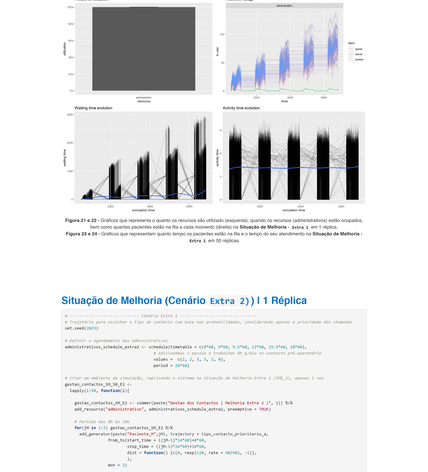


Figura 21 e 22 - Gráficos que representa o quanto os recursos são utilizados (esquerda), quanto os recursos (administrativos) estão ocupados, bem como quantas pacientes estão na fila a cada momento (direita) da **Situação de Melhoria - Extra 1**, em 1 réplica.

Figura 23 e 24 - Gráficos que representam quanto tempo os pacientes estão na fila e o tempo do seu atendimento na **Situação de Melhoria - Extra 1**, em 50 réplicas.

Situação de Melhoria (Cenário `Extra 2`)) I 1 Réplica

```
# ------------------------------ Cenário Extra 2 ------------------------------
# Trajetória para escolher o tipo de contacto com base nas probabilidades, considerando apenas a prioridade das chamadas
set.seed(2023)

# Defnir o agendamento dos administrativos
administrations_schedule_extra2 <- schedule(timetable = c(5*60, 9*60, 9.5*60, 12*60, 15.5*60, 18*60),
                                            # dois valores 1 pessoa e trabalhar de acordo no contacto pre-operatório
                                            values = c(0.5, 1, 0.5, 1, 0.5),
                                            period = 24*60)

# Criar um ambiente de simulação, replicando o sistema na Situação de Melhoria Extra 2 (SM_2), apenas 1 vez
gestao_contactos_SM_50_E2 <-
  lapply(1:50, function(i){

    gestao_contactos_SM_50 <- simmer(paste("Gestao dos Contactos | Melhoria Extra 2 ", i) )%>%
      add_resource("administration", administrations_schedule_extra2, preemptive = TRUE)

    # Periodo das 8h ás 14h
    for(j IN 1:5) gestao_contactos_SM_E2 %>%
      add_generator(paste("Paciente_M", j=j, trajectory = tipo_contacto_prioritario_a,
                          from_to(start_time = ((24-1)*5+7*60)*60,
                                  stop_time = ((24-1)*5+9*60)*60,
                                  dist = function(){c(0, rexp(119, rate = 60/60), -1)},
                                  ),
                          mon = 1)

    # Periodo das 14h ás 18h
    for(j IN 1:5) gestao_contactos_SM_E2 %>%
      add_generator(paste("Paciente_T", j=j, trajectory = tipo_contacto_prioritario_a,
                          from_to(start_time = ((24-1)*5+14*60)*60,
                                  stop_time = ((24-1)*5+18*60)*60,
                                  dist = function(){c(0, rexp(119, rate = 60/60), -1)},
                                  ),
                          mon = 1)

    # Periodo das 18h ás 20h
    for(j IN 1:5) gestao_contactos_SM_E2 %>%
      add_generator(paste("Paciente_N", j=j, trajectory = tipo_contacto_prioritario_a,
                          from_to(start_time = ((24-1)*5+18*60)*60,
                                  stop_time = ((24-1)*5+20*60)*60,
                                  dist = function(){c(0, rexp(119, rate = 20/60), -1)},
                                  ),
                          mon = 1)

    gestao_contactos_SM_E2 %>% run(until = 5*24*60) %>% invisible
})

# Métricas para a análise da eficiência (Exercício 2 | Cenário Extra 2) )
df <- as.data.frame(metrics_fun(gestao_contactos_SM_50_E2))
ftable_1 <- ftable(df)

ftable_1 <- ftable_1, hg = "#002060", part = "header")
ftable_1 <- color(ftable_1, color = "white", part = "header")
ftable_1 <- bold(ftable_1, bold = TRUE, part="header")
ftable_1 <- autofit(ftable_1)
ftable_1 %>%
  hline( i = "body", j= 1, j = 1) %>%
  hline(i = 1, part = "body", border = fp_border(color = "#002060", style = "dashed", width = 1)) %>%
  hline(i = "body", j= 4, j = 5) %>%
  hline(i = 1, part = "body", border = fp_border(color = "#002060", style = "dashed", width = 1)) %>%
  hline(part = "body", i= 1, j = 1) %>%
  hline(i = 5, part = "body", border = fp_border(color = "#002060", style = "dashed", width = 1)) %>%
  hline(part = "body", i= 15, j = 1) %>%
  hline(i = 12, part = "body", border = fp_border(color = "#002060", style = "solid", width = 1))
```

Métrica	Valor	%
Total de Pacientes	1993	
Total de Pacientes Atendidos	1582	79.6
Pacientes Totais Não Atendidos	401	20.4
Total de Chamadas	1667	
Chamadas Telefónicas Atendidas	1271	76.2
Chamadas Perdidas	396	23.8
Total de Mensagens	296	
Mensagens Respondidas	291	98.4
Mensagens Não Respondidas	5	1.6
Pacientes Consulta Presencial	157	10.0
Pacientes Consulta Telefónica	942	60.3
Pacientes Sem Necessidade	464	29.7
Tempo Médio dos Pacientes Atendidos [minutos]	**Média**	**DP**
Tempo de Espera Médio	25.67	7.3
Tempo de Espera Médio [Chamadas]	4.29	0.8
Tempo de Espera Médio [Mensagens]	146.79	45.8
Tempo de Serviço Médio	5.71	0.0
Tempo Total De Processo	30.22	7.4

Tabela 10 - Variáveis de análise da eficiência da 3ª Simulação - **Situação de Melhoria - Cenário** `Extra 2` | 1 Réplica.

```
gestao_contactos_SM_E1 %>% get_mon_resources() %>% plot(metric = "utilization")
gestao_contactos_SM_E1 %>% get_mon_resources() %>% plot(metric = "usage", steps=T)
gestao_contactos_SM_E1 %>% get_mon_arrivals() %>% plot(metric = "waiting_time")
gestao_contactos_SM_E1 %>% get_mon_arrivals() %>% plot(metric = "activity_time")
```

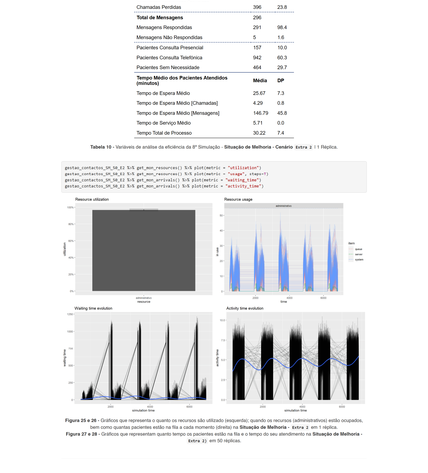


Figura 25 e 26 - Gráficos que representa o quanto os recursos são utilizados (esquerda), quanto os recursos (administrativos) estão ocupados, bem como quantas pacientes estão na fila a cada momento (direita) da **Situação de Melhoria - Extra 2**, em 1 réplica.

Figura 27 e 28 - Gráficos que representam quanto tempo os pacientes estão na fila e o tempo do seu atendimento na **Situação de Melhoria - Extra 2**, em 50 réplicas.

Resumo da Análise das Eficiências nas Simulações

```
# Tabela Geral dos Métricas para a análise da eficiência das várias situações
df <- as.data.frame(rbind(metrics_fun(gestao_contactos_SA_50), metrics_fun(gestao_contactos_SM_50a)[2:3],
                          metrics_fun(gestao_contactos_SM_50b)[2:3], metrics_fun(gestao_contactos_SM_50_E1)[2:3],
                          metrics_fun(gestao_contactos_SM_50_E2)[2:3]))
ftable_2 <- ftable(df)

ftable_2 <- tg(ftable_2, hg = "#002060", part = "header")
ftable_2 <- set_header_labels(ftable_2, values = c("Métrica", "SA", "", "SM aj", "", "SM b1", "", "SM | Extra 1", "", "SM | Extra 2", "") )
ftable_2 <- add_header_row(ftable_2, values = c("", "SA", "SM aj", "SM b1", "SM | Extra 1", "SM | Extra 2"), colwidths = c(1, 2, 2, 2, 2, 2)) %
>%
  bg(i = 1, j = 1:12, bg = "white", part = "header")
  style(i = 1, pr_t = fp_text_default(bold = T), color = "#002060", part = "header") %>%
  merge_at(i = 1, j = 2:3) %>% align(i = 1, j = 2:3, align = "center", part = "header") %>%
  merge_at(i = 1, j = 4:5) %>% align(i = 1, j = 4:5, align = "center", part = "header") %>%
  merge_at(i = 1, j = 6:7) %>% align(i = 1, j = 6:7, align = "center", part = "header") %>%
  merge_at(i = 1, j = 8:9) %>% align(i = 1, j = 8:9, align = "center", part = "header") %>%
  merge_at(i = 1, j = 10:11) %>% align(i = 1, j = 10:11, align = "center", part = "header")

ftable_2 %>%
  hline(part=part = "all", border = fp_border(color = "white", width = 1)) %>%
  hline(i = 1, part = "header", border = fp_border(color = "white", style = "solid", width = 1)) %>%
  hline(i = 2, part = "header", border = fp_border(color = "white", style = "solid", width = 1)) %>%
  hline(part = "body", i= 1, j = 1) %>%
  hline(i = 1, part = "body", border = fp_border(color = "#002060", style = "dashed", width = 1)) %>%
  hline(part = "body", i= 4, j = 1) %>%
  hline(i = 4, part = "body", border = fp_border(color = "#002060", style = "dashed", width = 1)) %>%
  hline(part = "body", i= 7, j = 1) %>%
  hline(i = 7, part = "body", border = fp_border(color = "#002060", style = "dashed", width = 1)) %>%
  hline(part = "body", i= 10, j = 1) %>%
  hline(i = 10, part = "body", border = fp_border(color = "#002060", style = "dashed", width = 1)) %>%
  hline(i = 13, part = "body", border = fp_border(color = "#002060", style = "solid", width = 1))
  vline(j = 1, part = "all", border = fp_border(color = "white", style = "solid", width = 10)) %>%
  vline(j = 2, part = "all", border = fp_border(color = "white", style = "solid", width = 10)) %>%
  vline(j = 3, part = "all", border = fp_border(color = "white", style = "solid", width = 10)) %>%
  vline(j = 4, part = "all", border = fp_border(color = "white", style = "solid", width = 10)) %>%
  vline(j = 5, part = "all", border = fp_border(color = "white", style = "solid", width = 10)) %>%
  vline(j = 6, part = "all", border = fp_border(color = "white", style = "solid", width = 10)) %>%
  vline(j = 14, part = "body", border = fp_border(color = "#FFFFFF", style = "solid", width = 10))
```

	SA		SM aj		SM b1		SM | Extra 1		SM | Extra 2	
Métrica	**Valor**	**%**	**Valor**	**%**	**Valor**	**%**	**Valor**	**%**	**Valor**	**%**
Total de Pacientes	1971		1971		1969		1966		1963	
Total de Pacientes Atendidos	1021	51.8	1313	66.6	1319	67.0	1365	69.5	1562	79.6
Pacientes Totais Não Atendidos	950	48.2	658	33.4	650	33.0	600	30.5	401	20.4
Total de Chamadas	1672		1676		1677		1673		1667	
Chamadas Telefónicas Atendidas	722	43.2	1111	66.3	1070	63.8	1148	68.6	1271	76.2
Chamadas Perdidas	950	56.8	566	33.7	607	36.2	525	31.4	396	23.8
Total de Mensagens	299		295		292		293		296	