

```
#####
##### Funcao Mares #####
#####

# Argumentos de entrada (local, periodo, mare)

mares <- function(localidade, periodo, mare)

{

#
# local = localidade para onde se deseja consultar a tabua de mares, de
# acordo com os locais disponiveis especificados no help da funcao
#
# periodo = Mes, trimestre ou semestre para o qual deseja a previsao de
# mares, de acordo com o especificado no help da funcao
#
# mare = Altura da mare desejada. Esse argumento eh opcional. Caso o
# argumento nao seja utilizado, a funcao usa o padrao de busca para mare de
# 0.3, valor geralmente utilizado para planejamento de expedicoes de coleta de
# animais da zona entre mare.
#

# Verificando os parametros

#

  locais.disp <- c("BELEM", "belem", "Belem", "bel", "CABEDEL0", "cabedelo",
"Cabedelo", "cab", "PARANAGUA", "paranagua", "Paranagua", "paran", "SUAPE",
"suape", "Suape", "sua", "RECIFE", "recife", "Recife", "rec", "NORONHA",
"F.noronha", "noronha", "f.nor", "LUIS.CORREIA",
"luis.correia", "Luis.Correia", "l.correia", "ILHA.FISCAL", "ilha.fiscal",
"Ilha.Fiscal", "i.fiscal", "ANGRA", "angra", "Angra", "angra.reis",
"IMBETIBA", "imbetiba", "Imbetiba", "Imbet", "NATAL", "natal", "Natal",
"nat", "MACEIO", "maceio", "Maceio", "mac", "RIO.GRANDE", "rio.grande",
"Rio.Grande", "r.grande", "FLORIANOPOLIS", "florianopolis", "Florianopolis",
"floripa", "ITAJAI", "itajai", "Itajai", "itaj", "SANTOS", "santos",
"Santos", "sant", "SAO.SEBASTIAO", "sao.sebastiao", "Sao.Sebastiao",
"s.sebastiao", "SERGIPE", "sergipe", "Sergipe", "serg", "SANTANA",
"santana", "Santana", "sant", "SALVADOR", "salvador", "Salvador", "salv",
"ILHEUS", "ilheus", "Ilheus", "ilh", "MUCURIBE", "mucuripe", "Mucuripe",
"muc", "PECEM", "pecem", "Pecem", "pec", "VITORIA", "vitoria", "Vitoria",
"vit", "UBU", "ubu", "Ubu", "SAO.LUIS", "sao.luis", "Sao.Luis",
"s.luis", "TRINDADE", "trindade", "trind", "AREIA.BRANCA", "areia.branca",
"Areia.Branca", "a.branca") #cria o vetor com as localidades disponiveis
para busca com possibilidade de letras maiusculas, minusculas, apenas
inicial maiuscula ou localidade abreviada
```

```
if (is.element(localidade, locais.disp) == F) #condicao para continuar a
funcao, soh continua se a localidade escolhida esta entre as localidades
pre-estabelecidas

{

  stop("Revise as localidades, uma ou mais nao correspondem a localidades
disponiveis para busca") #caso a localidade nao esteja entre as localidades
pre-estabelecidas, para a funcao com o aviso "Revise as localidades, uma ou
mais nao correspondem a localidades disponiveis para busca"

}

# O periodo especificado esta dentro esperado? Caso seja diferente do
esperado, retorna "Revise o periodo, nao se encaixa no esperado"

periodo.disp <- c("sem.1", "sem.2", "tri.1", "tri.2", "tri.3", "tri.4",
"jan", "fev", "mar", "abr", "jun", "jul", "ago", "set", "out", "nov", "dez")
#criando vetor com os periodos que podem ser utilizados na busca

if (is.element(periodo, periodo.disp) == FALSE) #condicao para continuar a
funcao, soh continua se o periodo escolhido esta entre os periodos
disponiveis para busca

{

  stop("Revise o periodo, nao se encaixa no esperado") #caso o periodo nao
esteja entre os periodos pre-estabelecidos, para a funcao com o aviso
"Revise o periodo, nao se encaixa no esperado"

}

# O argumento mare foi utilizado? Se nao, retorna "Nenhuma altura de mare
foi especificada, buscando mares iguais a 0.3"

if (missing(mare)) #condiciona para o uso do argumento mare, se nao foi
usado

{

  print("Nenhuma altura de mare foi especificada, buscando mares iguais a
0.3") #avisa para o usuario que o argumento mare nao foi usado e que por
isso sera o usado o valor padrao de 0.3
```

```
mare = "0.3" #configura o argumento mare para usar o valor padrao de 0.3

}
```

# 0 argumento mare foi utilizado, mas diferente das mares que ocorrem no Brasil, retorna "Altura de mare alem das registradas para o Brasil, revise"

```
d.mare <- c("-0.2","-0.1","0.0","0.1", "0.2", "0.3", "0.4", "0.5", "0.6",
"0.7", "0.8",
"0.9","1.0","1.1","1.2","1.3","1.4","1.5","1.6","1.7","1.8","1.9", "1.9",
"2.0", "2.1", "2.2", "2.3", "2.4", "2.5", "2.6", "2.7", "2.8", "2.9", "3.0",
"3.1", "3.2", "3.3", "3.4", "3.5", "3.6", "3.7", "3.8", "3.9", "4.0", "4.1",
"4.2", "4.3", "4.4", "4.5", "4.6", "4.7", "4.8", "4.9", "5.0", "5.1", "5.2",
"5.3", "5.4", "5.5", "5.6", "5.7", "5.8", "5.9", "6.0", "6.1", "6.2", "6.3",
"6.4", "6.5") #cria o vetor com as alturas de mares existentes no brasil
```

```
if (is.element(mare, d.mare)== FALSE) #condiciona para a variavel usada no
argumento mare, se a variavel estiver fora das mares especificadas no vetor
d.mare
```

```
{
```

```
  stop("Altura de mare alem das registradas para o Brasil, revise") #para a
funcao e retorna para o usuario a mensagem "Altura de mare alem das
registradas para o Brasil, revise"
```

```
}
```

# 1. Modificar a URL base para baixar o pdf da localidade solicitada

```
loc <- c("belem", "cabedelo", "paranagua_cais_oeste", "suape","recife",
"ilha_de_fernando_de_noronha_baia_de_santo_antonio", "luis_correia",
"rio_de_janeiro_ilha_fiscal", "angra_dos_reis",
"terminal_maritimo_de_imbetiba", "natal_capitania_dos_portos_do_rn",
"maceio", "rio_grande", "florianopolis", "itajai", "santos_torre_grande",
"sao_sebastiao", "capitania_dos_portos_de_sergipe",
"santana_cia._docas_de_santana", "salvador", "ilheus_malhado", "mucuripe",
"terminal_portuario_do_pecem", "vitoria_capitania_dos_portos_do_es",
```

```
"terminal_da_ponta_do_ubu", "sao_luis", "ilha_da_trindade",  
"areia_branca_termisa") #cria o vetor loc com o nome dos arquivos de  
localidade disponiveis para busca
```

```
loc <- as.data.frame(loc) #transforma o vetor loc em um data.frame pra  
facilitar o match entre esse vetor e o vetor localidade no loop que vem  
abaixo
```

```
localidade <- as.data.frame(localidade) #transforma a variavel fornecida no  
argumento localidade em data.frame pra facilitar o match entre esse vetor e  
o vetor loc no loop que vem abaixo
```

```
for (i in 1:nrow(localidade)) #cria um for loop com valores de i a cada  
numero de linhas do objeto localidade
```

```
{
```

```
    localidade$busca[i] <- grep(pattern= localidade[i,1], x=loc [,1],  
ignore.case = T, value = T) #cria uma nova coluna no data.frame localidade  
chamada busca e, a cada valor de i coloca nessa coluna na posicao i o  
resultado do grep. O grep usa a localidade na linha de posicao i, na coluna  
1 como padrao de busca para procurar a localidade correspondente em todas as  
linhas da coluna 1 do data.frame loc, ignorando a capitalizacao e tendo como  
resultado o string real, nao sua posicao
```

```
}
```

```
obj <- as.data.frame(NA) #cria o objeto obj, transforma ele em data.frame e  
e coloca NA
```

```
for (i in localidade$busca) #cria um loop para baixar as tabuas de cada  
localidade chamada, essa foi uma tentativa de utilizar mais de uma  
localidade na funcao, mas devido a passos posteriores ainda só é possível  
utilizar uma localidade
```

```
{

  obj[i] <- as.character(paste(i, "_2018.pdf", sep = "")) #coloca no obj na
posicao i o nome da localidade junto com o final _2018.pdf que esta presente
na url onde o pdf sera baixado

  locais <-
paste0("https://www.marinha.mil.br/chm/sites/www.marinha.mil.br.chm/files/da
dos_de_mare/", i, "_2018.pdf") # guarda no objeto locais a url que sera
utilizada para baixar a tabua em pdf

  tabua <- paste0(getwd(), "/", basename(locais)) #guarda no objeto tabua o
local onde a tabua de mares sera baixada, nesse caso o diretorio local do R

  download.file(locais, tabua) #faz o download da tabua de mares com base na
URL criada dois passos antes e guarda no local criado no passo anterior

}

# 2. Converter os arquivos .pdf baixados para .txt para facilitar a busca de
padrões

library(pdftools) #carrega o pacote pdftools necessario para a leitura de
pdfs no R

library(tidyr) #carrega o pacote tidyr necessario para separar o pdf em
colunas

for (i in 2:length(obj)) #cria o loop com i assumindo valores a cada posicao
dentro de obj (essa tambem foi uma tentativa de utilizar mais de uma
localidade na funcao)

{

  assign(paste("tabua",i, sep = ""), strsplit(x =
pdf_text(paste(obj[i])), split = c("\r\n"))) #cria o objeto tabua"i", que
sera preenchido pelo pdf lido pela funcao pdf_text e formatado pela funcao
strsplit utilizando como separador os caracteres "\r\n"

}
```

```
for (i in 1:length(tabua2)) # cria o loop com base no objeto criado no loop anterior, com i assumindo valores a cada elemento da lista
```

```
{  
  
    assign(paste("tabua",i, sep = "_"), tabua2[[i]][c(-1,-2,-3,-4,-5)])  
#cria tabua_"i" e guarda nela cada elemento da lista retirando as linhas de 1 a 5 que sao informacao irrelevantes para o uso da funcao e que atrapalham a formatacao das tabuas  
  
}
```

```
tabua_1 <- as.data.frame(tabua_1) #transforma tabua_1 em data.frame
```

```
tabua_2 <- as.data.frame(tabua_2) #transforma tabua_2 em data.frame
```

```
tabua_3 <- as.data.frame(tabua_3) #transforma tabua_3 em data.frame
```

```
##Formatando os dataframes criados no passo anterior. Sao necessarias condicionantes, pois diferentes localidades possuem pdfs com diferentes formatacoes
```

```
if(grepl("terminal_portuario_do_pecem", localidade$busca) |  
grepl("terminal_da_ponta_do_ubu",localidade$busca) == T) # cria condicao caso a localidade selecionada no argumento localidade seja pecem ou ubu
```

```
{  
  
cac <- tabua_1 %>% separate(tabua_1, sep = c(7, 18, 28, 39, 47, 56, 65, 75, 85, 95, 102, 111, 120, 130, 138), into = c("Janeiro1", "Janeiro3", "Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2", "Fevereiro4", "Março1", "Março3", "Março2", "Março4", "Abril1", "Abril3", "Abril2", "Abril4"),extra = "merge") #transforma tabua_1 em um dataframe com 16 colunas separadas de acordo com a posicao determinada em sep e nomeia as colunas como determinado em into, os caracteres extra ao final da tabela sao combinados na ultima coluna
```

```
cac2 <- tabua_2 %>% separate(tabua_2, sep = c(7,18, 28, 39, 46, 57, 64, 75, 84, 95, 102, 112, 120, 132, 138), into = c("Maio1", "Maio3","Maio2",
```

```
"Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1", "Julho3",
"Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2", "Agosto4"),extra =
"merge") #transforma tabua_2 em um dataframe com 16 colunas separadas de
acordo com a posicao determinada em sep e nomeia as colunas como determinado
em into, os caracteres extra ao final da tabela sao combinados na ultima
coluna
```

```
cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7, 18, 28, 39, 46, 56, 64, 74,
84, 93, 101, 111, 120, 130, 137), into = c("Setembro1", "Setembro3",
"Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4",
"Novembro1", "Novembro3", "Novembro2", "Novembro4", "Dezembro1", "Dezembro3",
"Dezembro2", "Dezembro4"),extra = "merge") #transforma tabua_3 em um
dataframe com 16 colunas separadas de acordo com a posicao determinada em
sep e nomeia as colunas como determinado em into, os caracteres extra ao
final da tabela sao combinados na ultima coluna
```

```
}
```

```
if(localidade$busca=="angra_dos_reis") #cria condicao caso a localidade
selecionada no argumento localidade seja angra dos reis
```

```
{
```

```
    cac <- tabua_1 %>% separate(tabua_1, sep = c(7, 18, 28, 39, 47, 57,
65, 75, 85, 95, 102, 113, 120, 134, 139), into = c("Janeiro1", "Janeiro3",
"Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2", "Fevereiro4",
"Março1", "Março3", "Março2", "Março4", "Abril1", "Abril3", "Abril2",
"Abril4"),extra = "merge") #transforma tabua_1 em um dataframe com 16
colunas separadas de acordo com a posicao determinada em sep e nomeia as
colunas como determinado em into, os caracteres extra ao final da tabela sao
combinados na ultima coluna
```

```
    cac2 <- tabua_2 %>% separate(tabua_2, sep = c(7,18, 28, 39, 48, 58,
66, 76, 84, 96, 102, 114, 120, 136, 139), into = c("Maio1", "Maio3","Maio2",
"Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1", "Julho3",
"Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2", "Agosto4"),extra =
"merge") #transforma tabua_2 em um dataframe com 16 colunas separadas de
acordo com a posicao determinada em sep e nomeia as colunas como determinado
em into, os caracteres extra ao final da tabela sao combinados na ultima
coluna
```

```
    cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7,18, 28, 39, 48, 58,
66, 76, 84, 97, 102, 115, 121, 136, 140), into = c("Setembro1", "Setembro3",
```

```
"Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4",  
"Novembro1", "Novembro3", "Novembro2", "Novembro4", "Dezembro1", "Dezembro3",  
"Dezembro2", "Dezembro4"), extra = "merge") #transforma tabua_3 em um  
dataframe com 16 colunas separadas de acordo com a posicao determinada em  
sep e nomeia as colunas como determinado em into, os caracteres extra ao  
final da tabela sao combinados na ultima coluna
```

```
}
```

```
if(grepl(pattern = "terminal_maritimo_de_imbetiba", x = localidade$busca) |  
grepl("luis_correia", localidade$busca)) # cria condicao caso a localidade  
selecionada no argumento localidade seja imbetiba ou luis correia
```

```
{  
  
  vetor <- c(7,18, 28, 39, 47, 57, 65, 75, 84, 95, 102, 113, 120, 131,  
138) #cria um vetor com os valores que serao utilizados como argumento para  
separacao em colunas dos objetos. Esses valores identificam a posicao em que  
a coluna ira terminar
```

```
  cac <- tabua_1 %>% separate(tabua_1, sep = vetor, into = c("Janeiro1",  
"Janeiro3", "Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2",  
"Fevereiro4", "Março1", "Março3", "Março2", "Março4", "Abril1", "Abril3",  
"Abril2", "Abril4"), extra = "merge") #transforma tabua_1 em um dataframe com  
16 colunas separadas de acordo com a posicao determinada em sep e nomeia as  
colunas como determinado em into, os caracteres extra ao final da tabela sao  
combinados na ultima coluna
```

```
  cac2 <- tabua_2 %>% separate(tabua_2, sep = c(7,18, 28, 38, 46, 56,  
64, 74, 84, 94, 100, 112, 120, 131, 138), into = c("Maio1", "Maio3", "Maio2",  
"Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1", "Julho3",  
"Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2", "Agosto4"), extra =  
"merge") #transforma tabua_2 em um dataframe com 16 colunas separadas de  
acordo com a posicao determinada em sep e nomeia as colunas como determinado  
em into, os caracteres extra ao final da tabela sao combinados na ultima  
coluna
```

```
  cac3 <- tabua_3 %>% separate(tabua_3, sep = vetor, into =  
c("Setembro1", "Setembro3", "Setembro2", "Setembro4", "Outubro1",  
"Outubro3", "Outubro2", "Outubro4", "Novembro1", "Novembro3", "Novembro2",  
"Novembro4", "Dezembro1", "Dezembro3", "Dezembro2", "Dezembro4"), extra =  
"merge") #transforma tabua_3 em um dataframe com 16 colunas separadas de  
acordo com a posicao determinada em sep e nomeia as colunas como determinado  
em into, os caracteres extra ao final da tabela sao combinados na ultima
```



```
coluna
```

```
}
```

```
if(grepl("maceio", localidade$busca) |  
grepl("natal_capitania_dos_portos_do_rn",localidade$busca) |  
grepl("areia_branca_termisa",localidade$busca) |  
grepl("recife",localidade$busca) == T) #cria condicao caso a localidade  
utilizada no argumento localidade seja maceio, natal, areia branca ou recife  
  
{
```

```
    vetor <-      c(7,18, 28, 39, 47, 57, 65, 75, 84, 94, 102, 112, 120,  
130, 137)  #cria um vetor com os valores que serao utilizados como  
argumento para separacao em colunas dos objetos. Esses valores identificam a  
posicao em que a coluna ira terminar
```

```
    cac <- tabua_1 %>% separate(tabua_1, sep = vetor, into = c("Janeiro1",  
"Janeiro3", "Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2",  
"Fevereiro4", "Marçol", "Março3", "Março2", "Março4", "Abril1", "Abril3",  
"Abril2", "Abril4"),extra = "merge") #transforma tabua_1 em um dataframe com  
16 colunas separadas de acordo com a posicao determinada em sep e nomeia as  
colunas como determinado em into, os caracteres extra ao final da tabela sao  
combinados na ultima coluna
```

```
    cac2 <- tabua_2 %>% separate(tabua_2, sep = vetor, into = c("Maio1",  
"Maio3","Maio2", "Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1",  
"Julho3", "Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2",  
"Agosto4"),extra = "merge") #transforma tabua_2 em um dataframe com 16  
colunas separadas de acordo com a posicao determinada em sep e nomeia as  
colunas como determinado em into, os caracteres extra ao final da tabela sao  
combinados na ultima coluna
```

```
    cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7,18, 28, 39, 46, 56,  
64, 74, 84, 93, 100, 111, 120, 130, 137), into = c("Setembro1", "Setembro3",  
"Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4",  
"Novembro1", "Novembro3","Novembro2", "Novembro4", "Dezembro1", "Dezembro3",  
"Dezembro2", "Dezembro4"),extra = "merge") #transforma tabua_3 em um  
dataframe com 16 colunas separadas de acordo com a posicao determinada em  
sep e nomeia as colunas como determinado em into, os caracteres extra ao  
final da tabela sao combinados na ultima coluna
```

```
}

if(grepl("rio_grande", localidade$busca) |
grepl("florianopolis",localidade$busca) |
grepl("sao_sebastiao",localidade$busca) == T) #cria condicao caso a
localidade utilizada no argumento localidade seja rio grande, florianopolis
ou sao sebastiao

{

    cac <- tabua_1 %>% separate(tabua_1, sep = c(7, 18, 28, 40, 47, 58,
65, 75, 85, 96, 102, 113, 120, 135, 140), into = c("Janeiro1", "Janeiro3",
"Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2", "Fevereiro4",
"Março1", "Março3", "Março2", "Março4", "Abril1", "Abril3", "Abril2",
"Abril4"),extra = "merge") #transforma tabua_1 em um dataframe com 16
colunas separadas de acordo com a posicao determinada em sep e nomeia as
colunas como determinado em into, os caracteres extra ao final da tabela sao
combinados na ultima coluna

    cac2 <- tabua_2 %>% separate(tabua_2, sep = c(7, 18, 28, 39, 47, 57,
65, 75, 85, 96, 102, 113, 120, 135, 140), into = c("Maio1", "Maio3","Maio2",
"Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1", "Julho3",
"Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2", "Agosto4"),extra =
"merge") #transforma tabua_3 em um dataframe com 16 colunas separadas de
acordo com a posicao determinada em sep e nomeia as colunas como determinado
em into, os caracteres extra ao final da tabela sao combinados na ultima
coluna

    cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7, 18, 28, 40, 47, 58,
65, 76, 85, 97, 102, 115, 121, 136, 140), into = c("Setembro1", "Setembro3",
"Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4",
"Novembro1", "Novembro3","Novembro2", "Novembro4", "Dezembro1", "Dezembro3",
"Dezembro2", "Dezembro4"),extra = "merge") #transforma tabua_3 em um
dataframe com 16 colunas separadas de acordo com a posicao determinada em
sep e nomeia as colunas como determinado em into, os caracteres extra ao
final da tabela sao combinados na ultima coluna

}
```

```
if(localidade$busca == "itajai") # cria condicao caso a localidade
selecionada no argumento localidade seja itajai

{

    vetor <- c(7, 18, 28, 39, 46, 56, 64, 74, 84, 94, 102, 113, 119, 134,
140) #cria um vetor com os valores que serao utilizados como argumento para
separacao em colunas dos objetos. Esses valores identificam a posicao em que
a coluna ira terminar

    cac <- tabua_1 %>% separate(tabua_1, sep = vetor, into = c("Janeiro1",
"Janeiro3", "Janeiro2", "Janeiro4", "Feveiro1", "Feveiro3", "Feveiro2",
"Feveiro4", "Marçol", "Março3", "Março2", "Março4", "Abrill", "Abril3",
"Abril2", "Abril4"),extra = "merge") #transforma tabua_1 em um dataframe com
16 colunas separadas de acordo com a posicao determinada em sep e nomeia as
colunas como determinado em into, os characters extra ao final da tabela sao
combinados na ultima coluna

    cac2 <- tabua_2 %>% separate(tabua_2, sep = vetor, into = c("Maio1",
"Maio3","Maio2", "Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1",
"Julho3", "Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2",
"Agosto4"),extra = "merge") #transforma tabua_2 em um dataframe com 16
colunas separadas de acordo com a posicao determinada em sep e nomeia as
colunas como determinado em into, os characters extra ao final da tabela sao
combinados na ultima coluna

    cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7, 18, 28, 39, 46, 56,
64, 74, 85, 96, 102, 113, 122, 134, 140), into = c("Setembro1", "Setembro3",
"Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4",
"Novembro1", "Novembro3","Novembro2", "Novembro4", "Dezembro1", "Dezembro3",
"Dezembro2", "Dezembro4"),extra = "merge") #transforma tabua_3 em um
dataframe com 16 colunas separadas de acordo com a posicao determinada em
sep e nomeia as colunas como determinado em into, os characters extra ao
final da tabela sao combinados na ultima coluna

}

if(localidade$busca == "paranagua_cais_oeste") # cria condicao caso a
localidade utilizada no argumento localidade seja paranagua

{
```

```
cac <- tabua_1 %>% separate(tabua_1, sep = c(7, 18, 28, 39, 46, 56, 64, 75, 84, 94, 102, 113, 119, 135, 140), into = c("Janeiro1", "Janeiro3", "Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2", "Fevereiro4", "Março1", "Março3", "Março2", "Março4", "Abril1", "Abril3", "Abril2", "Abril4"), extra = "merge") #transforma tabua_1 em um dataframe com 16 colunas separadas de acordo com a posicao determinada em sep e nomeia as colunas como determinado em into, os caracteres extra ao final da tabela sao combinados na ultima coluna
```

```
cac2 <- tabua_2 %>% separate(tabua_2, sep = c(7, 18, 28, 39, 46, 56, 64, 74, 84, 94, 102, 113, 119, 134, 140), into = c("Maio1", "Maio3", "Maio2", "Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1", "Julho3", "Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2", "Agosto4"), extra = "merge") #transforma tabua_2 em um dataframe com 16 colunas separadas de acordo com a posicao determinada em sep e nomeia as colunas como determinado em into, os caracteres extra ao final da tabela sao combinados na ultima coluna
```

```
cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7, 18, 28, 39, 46, 58, 64, 76, 85, 97, 102, 115, 122, 137, 141), into = c("Setembro1", "Setembro3", "Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4", "Novembro1", "Novembro3", "Novembro2", "Novembro4", "Dezembro1", "Dezembro3", "Dezembro2", "Dezembro4"), extra = "merge") #transforma tabua_3 em um dataframe com 16 colunas separadas de acordo com a posicao determinada em sep e nomeia as colunas como determinado em into, os caracteres extra ao final da tabela sao combinados na ultima coluna
```

```
}
```

```
if(localidade$busca == "santos_torre_grande") # cria condicao caso a localidade utilizada no argumento localidade seja santos
```

```
{
```

```
  cac <- tabua_1 %>% separate(tabua_1, sep = c(7, 18, 28, 39, 47, 57, 65, 75, 85, 95, 102, 115, 121, 133, 140), into = c("Janeiro1", "Janeiro3", "Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2", "Fevereiro4", "Março1", "Março3", "Março2", "Março4", "Abril1", "Abril3", "Abril2", "Abril4"), extra = "merge") #transforma tabua_1 em um dataframe com 16 colunas separadas de acordo com a posicao determinada em sep e nomeia as colunas como determinado em into, os caracteres extra ao final da tabela sao combinados na ultima coluna
```

```

cac2 <- tabua_2 %>% separate(tabua_2, sep = c(7, 18, 28, 39, 47, 57,
65, 75, 85, 95, 102, 115, 121, 136, 140), into = c("Maio1", "Maio3", "Maio2",
"Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1", "Julho3",
"Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2", "Agosto4"), extra =
"merge") #transforma tabua_1 em um dataframe com 16 colunas separadas de
acordo com a posicao determinada em sep e nomeia as colunas como determinado
em into, os caracteres extra ao final da tabela sao combinados na ultima
coluna

```

```

cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7, 18, 28, 39, 47, 57,
65, 75, 85, 96, 102, 115, 121, 133, 140), into = c("Setembro1", "Setembro3",
"Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4",
"Novembro1", "Novembro3", "Novembro2", "Novembro4", "Dezembro1", "Dezembro3",
"Dezembro2", "Dezembro4"), extra = "merge") #transforma tabua_1 em um
dataframe com 16 colunas separadas de acordo com a posicao determinada em
sep e nomeia as colunas como determinado em into, os caracteres extra ao
final da tabela sao combinados na ultima coluna

```

```

}

```

```

if(grepl("capitania_dos_portos_de_sergipe",localidade$busca) |
grepl("ilheus_malhado",localidade$busca) == T) #cria condicao caso a
localidade utilizada no argumento localidade seja sergipe ou ilheus

```

```

{

```

```

cac <- tabua_1 %>% separate(tabua_1, sep = c(7,18, 28, 39, 47, 57, 65,
75, 84, 94, 102, 112, 120, 130, 137), into = c("Janeiro1", "Janeiro3",
"Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2", "Fevereiro4",
"Março1", "Março3", "Março2", "Março4", "Abril1", "Abril3", "Abril2",
"Abril4"), extra = "merge") #transforma tabua_1 em um dataframe com 16
colunas separadas de acordo com a posicao determinada em sep e nomeia as
colunas como determinado em into, os caracteres extra ao final da tabela sao
combinados na ultima coluna

```

```

cac2 <- tabua_2 %>% separate(tabua_2, sep = c(7,18, 28, 39, 47, 57,
65, 75, 84, 94, 102, 112, 120, 131, 137), into = c("Maio1", "Maio3", "Maio2",
"Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1", "Julho3",

```

```
"Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2", "Agosto4"),extra =  
"merge") #transforma tabua_3 em um dataframe com 16 colunas separadas de  
acordo com a posicao determinada em sep e nomeia as colunas como determinado  
em into, os caracteres extra ao final da tabela sao combinados na ultima  
coluna
```

```
    cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7,18, 28, 39, 46, 56,  
64, 74, 84, 93, 100, 111, 120, 130, 137), into = c("Setembro1", "Setembro3",  
"Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4",  
"Novembro1", "Novembro3","Novembro2", "Novembro4", "Dezembro1", "Dezembro3",  
"Dezembro2", "Dezembro4"),extra = "merge") #transforma tabua_3 em um  
dataframe com 16 colunas separadas de acordo com a posicao determinada em  
sep e nomeia as colunas como determinado em into, os caracteres extra ao  
final da tabela sao combinados na ultima coluna  
  
}
```

```
if(grepl("santana_cia._docas_de_santana",localidade$busca) |  
grepl("salvador",localidade$busca) == T) # cria condicao caso a localidade  
utilizada no argumento localidade seja santana ou salvador  
  
{
```

```
    cac <- tabua_1 %>% separate(tabua_1, sep = c(7, 18, 28, 39, 47, 56,  
65, 75, 85, 95, 102, 111, 120, 130, 138), into = c("Janeiro1", "Janeiro3",  
"Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2", "Fevereiro4",  
"Marçol", "Março3", "Março2", "Março4", "Abril1", "Abril3", "Abril2",  
"Abril4"),extra = "merge") #transforma tabua_1 em um dataframe com 16  
colunas separadas de acordo com a posicao determinada em sep e nomeia as  
colunas como determinado em into, os caracteres extra ao final da tabela sao  
combinados na ultima coluna
```

```
    cac2 <- tabua_2 %>% separate(tabua_2, sep = c(7,18, 28, 39, 46, 56,  
64, 74, 84, 94, 102, 111, 120, 130, 137), into = c("Maio1", "Maio3","Maio2",  
"Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1", "Julho3",  
"Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2", "Agosto4"),extra =  
"merge") #transforma tabua_2 em um dataframe com 16 colunas separadas de  
acordo com a posicao determinada em sep e nomeia as colunas como determinado  
em into, os caracteres extra ao final da tabela sao combinados na ultima  
coluna
```

```

cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7, 18, 28, 39, 46, 56,
64, 74, 84, 93, 101, 111, 120, 131, 137), into = c("Setembro1", "Setembro3",
"Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4",
"Novembro1", "Novembro3", "Novembro2", "Novembro4", "Dezembro1", "Dezembro3",
"Dezembro2", "Dezembro4"), extra = "merge") #transforma tabua_3 em um
dataframe com 16 colunas separadas de acordo com a posicao determinada em
sep e nomeia as colunas como determinado em into, os caracteres extra ao
final da tabela sao combinados na ultima coluna

}

```

```

if(localidade$busca=="mucuripe") # cria condicao caso a localidade utilizada
no argumento localidade seja mucuripe

```

```

{

    vetor <- c(7, 18, 28, 39, 47, 57, 65, 75, 85, 95, 102, 113, 120, 132,
139) #cria um vetor com os valores que serao utilizados como argumento para
separacao em colunas dos objetos. Esses valores identificam a posicao em que
a coluna ira terminar

```

```

cac <- tabua_1 %>% separate(tabua_1, sep = vetor, into = c("Janeiro1",
"Janeiro3", "Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2",
"Fevereiro4", "Março1", "Março3", "Março2", "Março4", "Abril1", "Abril3",
"Abril2", "Abril4"), extra = "merge") #transforma tabua_1 em um dataframe com
16 colunas separadas de acordo com a posicao determinada em sep e nomeia as
colunas como determinado em into, os caracteres extra ao final da tabela sao
combinados na ultima coluna

```

```

cac2 <- tabua_2 %>% separate(tabua_2, sep = vetor, into = c("Maio1",
"Maio3", "Maio2", "Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1",
"Julho3", "Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2",
"Agosto4"), extra = "merge") #transforma tabua_2 em um dataframe com 16
colunas separadas de acordo com a posicao determinada em sep e nomeia as
colunas como determinado em into, os caracteres extra ao final da tabela sao
combinados na ultima coluna

```

```

cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7, 18, 28, 40, 48, 58,
66, 76, 84, 97, 102, 115, 121, 134, 140), into = c("Setembro1", "Setembro3",
"Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4",
"Novembro1", "Novembro3", "Novembro2", "Novembro4", "Dezembro1", "Dezembro3",
"Dezembro2", "Dezembro4"), extra = "merge") #transforma tabua_3 em um
dataframe com 16 colunas separadas de acordo com a posicao determinada em

```

sep e nomeia as colunas como determinado em into, os caracteres extra ao final da tabela são combinados na última coluna

```
}

if(localidade$busca == "vitoria_capitania_dos_portos_do_es") #cria condicao
caso a localidade utilizada no argumento localidade seja vitoria

{

    cac <- tabua_1 %>% separate(tabua_1, sep = c(7, 18, 28, 39, 46, 56,
64, 74, 84, 94, 101, 112, 116, 130, 137), into = c("Janeiro1", "Janeiro3",
"Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2", "Fevereiro4",
"Março1", "Março3", "Março2", "Março4", "Abril1", "Abril3", "Abril2",
"Abril4"), extra = "merge") #transforma tabua_1 em um dataframe com 16
colunas separadas de acordo com a posicao determinada em sep e nomeia as
colunas como determinado em into, os caracteres extra ao final da tabela são
combinados na última coluna

    cac2 <- tabua_2 %>% separate(tabua_2, sep = c(7, 18, 28, 39, 46, 56,
64, 74, 84, 94, 101, 112, 116, 131, 137), into = c("Maio1", "Maio3", "Maio2",
"Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1", "Julho3",
"Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2", "Agosto4"), extra =
"merge") #transforma tabua_2 em um dataframe com 16 colunas separadas de
acordo com a posicao determinada em sep e nomeia as colunas como determinado
em into, os caracteres extra ao final da tabela são combinados na última
coluna

    cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7, 18, 28, 39, 46, 56,
64, 74, 84, 93, 101, 111, 116, 130, 137), into = c("Setembro1", "Setembro3",
"Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4",
"Novembro1", "Novembro3", "Novembro2", "Novembro4", "Dezembro1", "Dezembro3",
"Dezembro2", "Dezembro4"), extra = "merge") #transforma tabua_3 em um
dataframe com 16 colunas separadas de acordo com a posicao determinada em
sep e nomeia as colunas como determinado em into, os caracteres extra ao
final da tabela são combinados na última coluna

}
```



```
if(localidade$busca == "sao_luis") # cria condicao caso a localidade
utilizada no argumento localidade seja sao luis

{

    cac <- tabua_1 %>% separate(tabua_1, sep = c(7, 18, 28, 39, 47, 56,
65, 75, 85, 94, 102, 111, 120, 130, 137), into = c("Janeiro1", "Janeiro3",
"Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2", "Fevereiro4",
"Março1", "Março3", "Março2", "Março4", "Abril1", "Abril3", "Abril2",
"Abril4"),extra = "merge") #transforma tabua_1 em um dataframe com 16
colunas separadas de acordo com a posicao determinada em sep e nomeia as
colunas como determinado em into, os caracteres extra ao final da tabela sao
combinados na ultima coluna

    cac2 <- tabua_2 %>% separate(tabua_2, sep = c(7,18, 28, 39, 46, 57,
64, 75, 84, 95, 102, 112, 120, 132, 138), into = c("Mai1", "Mai3","Mai2",
"Mai4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1", "Julho3",
"Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2", "Agosto4"),extra =
"merge") #transforma tabua_2 em um dataframe com 16 colunas separadas de
acordo com a posicao determinada em sep e nomeia as colunas como determinado
em into, os caracteres extra ao final da tabela sao combinados na ultima
coluna

    cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7, 18, 28, 39, 46, 56,
64, 74, 84, 93, 101, 111, 116, 130, 137), into = c("Setembro1", "Setembro3",
"Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4",
"Novembro1", "Novembro3","Novembro2", "Novembro4", "Dezembro1", "Dezembro3",
"Dezembro2", "Dezembro4"),extra = "merge") #transforma tabua_3 em um
dataframe com 16 colunas separadas de acordo com a posicao determinada em
sep e nomeia as colunas como determinado em into, os caracteres extra ao
final da tabela sao combinados na ultima coluna

}

if(grepl("ilha_da_trindade",localidade$busca) |
grepl("cabedelo",localidade$busca) == T) #cria condicao caso a localidade
utilizada no argumento localidade seja trindade ou cabedelo

{
```

```
cac <- tabua_1 %>% separate(tabua_1, sep = c(7, 18, 28, 39, 48, 57, 65, 75, 85, 95, 102, 112, 120, 131, 137), into = c("Janeiro1", "Janeiro3", "Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2", "Fevereiro4", "Março1", "Março3", "Março2", "Março4", "Abril1", "Abril3", "Abril2", "Abril4"), extra = "merge") #transforma tabua_1 em um dataframe com 16 colunas separadas de acordo com a posicao determinada em sep e nomeia as colunas como determinado em into, os caracteres extra ao final da tabela sao combinados na ultima coluna
```

```
cac2 <- tabua_2 %>% separate(tabua_2, sep = c(7, 18, 28, 39, 46, 56, 64, 74, 84, 93, 101, 111, 120, 130, 137), into = c("Maio1", "Maio3", "Maio2", "Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1", "Julho3", "Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2", "Agosto4"), extra = "merge") #transforma tabua_2 em um dataframe com 16 colunas separadas de acordo com a posicao determinada em sep e nomeia as colunas como determinado em into, os caracteres extra ao final da tabela sao combinados na ultima coluna
```

```
cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7, 18, 28, 39, 46, 56, 64, 74, 84, 93, 101, 111, 116, 130, 137), into = c("Setembro1", "Setembro3", "Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4", "Novembro1", "Novembro3", "Novembro2", "Novembro4", "Dezembro1", "Dezembro3", "Dezembro2", "Dezembro4"), extra = "merge") #transforma tabua_3 em um dataframe com 16 colunas separadas de acordo com a posicao determinada em sep e nomeia as colunas como determinado em into, os caracteres extra ao final da tabela sao combinados na ultima coluna
```

```
}
```

```
if(grepl("belem",localidade$busca) | grepl("suape",localidade$busca) == T)  
#cria condicao caso a localidade utilizada no argumento localidade seja  
belem ou suape
```

```
{
```

```
cac <- tabua_1 %>% separate(tabua_1, sep = c(7, 18, 28, 39, 48, 57, 65, 75, 85, 95, 102, 112, 120, 131, 137), into = c("Janeiro1", "Janeiro3", "Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2", "Fevereiro4", "Março1", "Março3", "Março2", "Março4", "Abril1", "Abril3", "Abril2", "Abril4"), extra = "merge") #transforma tabua_1 em um dataframe com 16 colunas separadas de acordo com a posicao determinada em sep e nomeia as colunas como determinado em into, os caracteres extra ao final da tabela sao
```

combinados na ultima coluna

```
cac2 <- tabua_2 %>% separate(tabua_2, sep = c(7,18, 28, 39, 46, 56,
64, 74, 84, 93, 101, 111, 120, 130, 137), into = c("Maio1", "Maio3","Maio2",
"Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1", "Julho3",
"Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2", "Agosto4"),extra =
"merge") #transforma tabua_2 em um dataframe com 16 colunas separadas de
acordo com a posicao determinada em sep e nomeia as colunas como determinado
em into, os caracteres extra ao final da tabela sao combinados na ultima
coluna
```

```
cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7, 18, 28, 39, 46, 56,
64, 74, 84, 93, 101, 111, 116, 131, 137), into = c("Setembro1", "Setembro3",
"Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4",
"Novembro1", "Novembro3","Novembro2", "Novembro4", "Dezembro1", "Dezembro3",
"Dezembro2", "Dezembro4"),extra = "merge") #transforma tabua_3 em um
dataframe com 16 colunas separadas de acordo com a posicao determinada em
sep e nomeia as colunas como determinado em into, os caracteres extra ao
final da tabela sao combinados na ultima coluna
```

```
}
```

```
if(localidade$busca == "ilha_de_fernando_de_noronha_baia_de_santo_antonio")
#cria condicao caso a localidade utilizada no argumento localidade seja
fernando de noronha
```

```
{
```

```
cac <- tabua_1 %>% separate(tabua_1, sep = c(7, 18, 28, 39, 48, 57,
65, 75, 85, 95, 102, 112, 120, 131, 137), into = c("Janeiro1", "Janeiro3",
"Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2", "Fevereiro4",
"Março1", "Março3", "Março2", "Março4", "Abril1", "Abril3", "Abril2",
"Abril4"),extra = "merge") #transforma tabua_1 em um dataframe com 16
colunas separadas de acordo com a posicao determinada em sep e nomeia as
colunas como determinado em into, os caracteres extra ao final da tabela sao
combinados na ultima coluna
```

```
cac2 <- tabua_2 %>% separate(tabua_2, sep = c(7,18, 28, 39, 46, 56,
64, 74, 84, 93, 101, 111, 120, 129, 136), into = c("Maio1", "Maio3","Maio2",
"Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1", "Julho3",
"Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2", "Agosto4"),extra =
```

```
"merge") #transforma tabua_2 em um dataframe com 16 colunas separadas de  
acordo com a posicao determinada em sep e nomeia as colunas como determinado  
em into, os caracteres extra ao final da tabela sao combinados na ultima  
coluna
```

```
    cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7, 18, 28, 39, 46, 56,  
64, 74, 84, 93, 101, 111, 116, 131, 136), into = c("Setembro1", "Setembro3",  
"Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4",  
"Novembro1", "Novembro3", "Novembro2", "Novembro4", "Dezembro1", "Dezembro3",  
"Dezembro2", "Dezembro4"), extra = "merge") #transforma tabua_3 em um  
dataframe com 16 colunas separadas de acordo com a posicao determinada em  
sep e nomeia as colunas como determinado em into, os caracteres extra ao  
final da tabela sao combinados na ultima coluna
```

```
}
```

```
if(localidade$busca == "rio_de_janeiro_ilha_fiscal") #cria condicao caso a  
localidade definida no argumento localidade seja rio de janeiro
```

```
{
```

```
    cac <- tabua_1 %>% separate(tabua_1, sep = c(7, 18, 28, 39, 47, 56,  
65, 75, 85, 95, 102, 112, 122, 133, 138), into = c("Janeiro1", "Janeiro3",  
"Janeiro2", "Janeiro4", "Fevereiro1", "Fevereiro3", "Fevereiro2", "Fevereiro4",  
"Marçol", "Março3", "Março2", "Março4", "Abril1", "Abril3", "Abril2",  
"Abril4"), extra = "merge") #transforma tabua_1 em um dataframe com 16  
colunas separadas de acordo com a posicao determinada em sep e nomeia as  
colunas como determinado em into, os caracteres extra ao final da tabela sao  
combinados na ultima coluna
```

```
    cac2 <- tabua_2 %>% separate(tabua_2, sep = c(7, 18, 28, 39, 46, 56,  
64, 74, 84, 93, 101, 111, 120, 131, 136), into = c("Maio1", "Maio3", "Maio2",  
"Maio4", "Junho1", "Junho3", "Junho2", "Junho4", "Julho1", "Julho3",  
"Julho2", "Julho4", "Agosto1", "Agosto3", "Agosto2", "Agosto4"), extra =  
"merge") #transforma tabua_2 em um dataframe com 16 colunas separadas de  
acordo com a posicao determinada em sep e nomeia as colunas como determinado  
em into, os caracteres extra ao final da tabela sao combinados na ultima  
coluna
```

```
cac3 <- tabua_3 %>% separate(tabua_3, sep = c(7, 18, 28, 40, 46, 56,
64, 75, 84, 96, 101, 114, 120, 135, 140), into = c("Setembro1", "Setembro3",
"Setembro2", "Setembro4", "Outubro1", "Outubro3", "Outubro2", "Outubro4",
"Novembro1", "Novembro3", "Novembro2", "Novembro4", "Dezembro1", "Dezembro3",
"Dezembro2", "Dezembro4"), extra = "merge") #transforma tabua_3 em um
dataframe com 16 colunas separadas de acordo com a posicao determinada em
sep e nomeia as colunas como determinado em into, os caracteres extra ao
final da tabela sao combinados na ultima coluna
```

```
}
```

```
cac[c(grep("01", cac$Janeiro1)-1):c(grep("02", cac$Janeiro1)-2),c(1,5,9,13)]
<- "01" #encontra dia 01 e preenche com 01 da celula anterior ate duas
celulas antes do dia 02 (esse eh o intervalo correspondente ao dia no
dataframe)
```

```
cac[c(grep("02", cac$Janeiro1)-1):c(grep("03", cac$Janeiro1)-2),c(1,5,9,13)]
<- "02" #encontra dia 02 e preenche com 02 da celula anterior ate duas
celulas antes do dia 03 (esse eh o intervalo correspondente ao dia no
dataframe)
```

```
cac[c(grep("03", cac$Janeiro1)-1):c(grep("04", cac$Janeiro1)-2),c(1,5,9,13)]
<- "03" #encontra dia 03 e preenche com 03 da celula anterior ate duas
celulas antes do dia 03 (esse eh o intervalo correspondente ao dia no
dataframe)
```

```
cac[c(grep("04", cac$Janeiro1)-1):c(grep("05", cac$Janeiro1)-2),c(1,5,9,13)]
<- "04" #encontra dia 04 e preenche com 04 da celula anterior ate duas
celulas antes do dia 05 (esse eh o intervalo correspondente ao dia no
dataframe)
```

```
cac[c(grep("05", cac$Janeiro1)-1):c(grep("06", cac$Janeiro1)-2),c(1,5,9,13)]
<- "05" #encontra dia 05 e preenche com 05 da celula anterior ate duas
celulas antes do dia 06 (esse eh o intervalo correspondente ao dia no
dataframe)
```

```
cac[c(grep("06", cac$Janeiro1)-1):c(grep("07", cac$Janeiro1)-2),c(1,5,9,13)]
<- "06" #encontra dia 06 e preenche com 06 da celula anterior ate duas
celulas antes do dia 07 (esse eh o intervalo correspondente ao dia no
dataframe)
```

```
cac[c(grep("07", cac$Janeiro1)-1):c(grep("08", cac$Janeiro1)-2),c(1,5,9,13)]
<- "07" #encontra dia 07 e preenche com 07 da celula anterior ate duas
celulas antes do dia 08 (esse eh o intervalo correspondente ao dia no
dataframe)
```

```
cac[c(grep("08", cac$Janeiro1)-1):c(grep("09", cac$Janeiro1)-2),c(1,5,9,13)]
```

```
<- "08" #encontra dia 08 e preenche com 08 da celula anterior ate duas
celulas antes do dia 09 (esse eh o intervalo correspondente ao dia no
dataframe)

cac[c(grep("09", cac$Janeiro1)-1):c(grep("10", cac$Janeiro1)-2),c(1,5,9,13)]
<- "09" #encontra dia 09 e preenche com 09 da celula anterior ate duas
celulas antes do dia 10 (esse eh o intervalo correspondente ao dia no
dataframe)

cac[c(grep("10", cac$Janeiro1)-1):c(grep("11", cac$Janeiro1)-2),c(1,5,9,13)]
<- "10" #encontra dia 10 e preenche com 10 da celula anterior ate duas
celulas antes do dia 11 (esse eh o intervalo correspondente ao dia no
dataframe)

cac[c(grep("11", cac$Janeiro1)-1):c(grep("12", cac$Janeiro1)-2),c(1,5,9,13)]
<- "11" #encontra dia 11 e preenche com 11 da celula anterior ate duas
celulas antes do dia 12 (esse eh o intervalo correspondente ao dia no
dataframe)

cac[c(grep("12", cac$Janeiro1)-1):c(grep("13", cac$Janeiro1)-2),c(1,5,9,13)]
<- "12" #encontra dia 12 e preenche com 12 da celula anterior ate duas
celulas antes do dia 13 (esse eh o intervalo correspondente ao dia no
dataframe)

cac[c(grep("13", cac$Janeiro1)-1):c(grep("14", cac$Janeiro1)-2),c(1,5,9,13)]
<- "13" #encontra dia 13 e preenche com 13 da celula anterior ate duas
celulas antes do dia 14 (esse eh o intervalo correspondente ao dia no
dataframe)

cac[c(grep("14", cac$Janeiro1)-1):c(grep("15", cac$Janeiro1)-2),c(1,5,9,13)]
<- "14" #encontra dia 14 e preenche com 14 da celula anterior ate duas
celulas antes do dia 15 (esse eh o intervalo correspondente ao dia no
dataframe)

cac[c(grep("15", cac$Janeiro1)-1):c(grep("16", cac$Janeiro1)-2),c(1,5,9,13)]
<- "15" #encontra dia 15 e preenche com 15 da celula anterior ate duas
celulas antes do dia 16 (esse eh o intervalo correspondente ao dia no
dataframe)

cac[c(grep("16", cac$Janeiro1)-1):nrow(cac),1] <- "16" #encontra dia 16 e
preenche com 16 da celula anterior ate o final das linhas do objeto (esse eh
o intervalo correspondente ao dia no dataframe)

cac[c(grep("17", cac$Janeiro2)-1):c(grep("18",
cac$Janeiro2)-2),c(3,7,11,15)] <- "17" #encontra dia 17 e preenche com 17 da
celula anterior ate duas celulas antes do dia 18 (esse eh o intervalo
correspondente ao dia no dataframe)
```

```
cac[c(grep("18", cac$Janeiro2)-1):c(grep("19",  
cac$Janeiro2)-2),c(3,7,11,15)] <- "18" #encontra dia 18 e preenche com 18 da  
celula anterior ate duas celulas antes do dia 19 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac[c(grep("19", cac$Janeiro2)-1):c(grep("20",  
cac$Janeiro2)-2),c(3,7,11,15)] <- "19" #encontra dia 19 e preenche com 19 da  
celula anterior ate duas celulas antes do dia 20 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac[c(grep("20", cac$Janeiro2)-1):c(grep("21",  
cac$Janeiro2)-2),c(3,7,11,15)] <- "20" #encontra dia 20 e preenche com 20 da  
celula anterior ate duas celulas antes do dia 21 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac[c(grep("21", cac$Janeiro2)-1):c(grep("22",  
cac$Janeiro2)-2),c(3,7,11,15)] <- "21" #encontra dia 21 e preenche com 21 da  
celula anterior ate duas celulas antes do dia 22 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac[c(grep("22", cac$Janeiro2)-1):c(grep("23",  
cac$Janeiro2)-2),c(3,7,11,15)] <- "22" #encontra dia 22 e preenche com 22 da  
celula anterior ate duas celulas antes do dia 23 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac[c(grep("23", cac$Janeiro2)-1):c(grep("24",  
cac$Janeiro2)-2),c(3,7,11,15)] <- "23" #encontra dia 23 e preenche com 23 da  
celula anterior ate duas celulas antes do dia 24 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac[c(grep("24", cac$Janeiro2)-1):c(grep("25",  
cac$Janeiro2)-2),c(3,7,11,15)] <- "24" #encontra dia 24 e preenche com 24 da  
celula anterior ate duas celulas antes do dia 25 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac[c(grep("25", cac$Janeiro2)-1):c(grep("26",  
cac$Janeiro2)-2),c(3,7,11,15)] <- "25" #encontra dia 25 e preenche com 25 da  
celula anterior ate duas celulas antes do dia 26 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac[c(grep("26", cac$Janeiro2)-1):c(grep("27",  
cac$Janeiro2)-2),c(3,7,11,15)] <- "26" #encontra dia 26 e preenche com 26 da  
celula anterior ate duas celulas antes do dia 27 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac[c(grep("27", cac$Janeiro2)-1):c(grep("28",  
cac$Janeiro2)-2),c(3,7,11,15)] <- "27" #encontra dia 27 e preenche com 27 da  
celula anterior ate duas celulas antes do dia 28 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac[c(grep("28", cac$Janeiro2)-1):c(grep("29",
```

```
cac$Janeiro2)-2),c(3,7,11,15)] <- "28" #encontra dia 28 e preenche com 28 da  
celula anterior ate duas celulas antes do dia 29 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac[c(grep("29", cac$Janeiro2)-1):c(grep("30",  
cac$Janeiro2)-2),c(3,7,11,15)] <- "29" #encontra dia 29 e preenche com 29 da  
celula anterior ate duas celulas antes do dia 30 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac[c(grep("30", cac$Janeiro2)-1):c(grep("31",  
cac$Janeiro2)-2),c(3,7,11,15)] <- "30" #encontra dia 30 e preenche com 30 da  
celula anterior ate duas celulas antes do dia 31 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac[c(grep("31", cac$Janeiro2)-1):nrow(cac),c(3,7,11,15)] <- "31" #encontra  
dia 31 e preenche com 31 da celula anterior ate o final das linhas do objeto  
(esse eh o intervalo correspondente ao dia no dataframe)
```

```
cac2[c(grep("01", cac2$Maio1)-1):c(grep("02", cac2$Maio1)-2),c(1,5,9,13)] <-  
"01" #encontra dia 01 e preenche com 01 da celula anterior ate duas celulas  
antes do dia 02 (esse eh o intervalo correspondente ao dia no dataframe)
```

```
cac2[c(grep("02", cac2$Maio1)-1):c(grep("03", cac2$Maio1)-2),c(1,5,9,13)] <-  
"02" #encontra dia 02 e preenche com 02 da celula anterior ate duas celulas  
antes do dia 03 (esse eh o intervalo correspondente ao dia no dataframe)
```

```
cac2[c(grep("03", cac2$Maio1)-1):c(grep("04", cac2$Maio1)-2),c(1,5,9,13)] <-  
"03" #encontra dia 03 e preenche com 03 da celula anterior ate duas celulas  
antes do dia 04 (esse eh o intervalo correspondente ao dia no dataframe)
```

```
cac2[c(grep("04", cac2$Maio1)-1):c(grep("05", cac2$Maio1)-2),c(1,5,9,13)] <-  
"04" #encontra dia 04 e preenche com 04 da celula anterior ate duas celulas  
antes do dia 05 (esse eh o intervalo correspondente ao dia no dataframe)
```

```
cac2[c(grep("05", cac2$Maio1)-1):c(grep("06", cac2$Maio1)-2),c(1,5,9,13)] <-  
"05" #encontra dia 05 e preenche com 05 da celula anterior ate duas celulas  
antes do dia 06 (esse eh o intervalo correspondente ao dia no dataframe)
```

```
cac2[c(grep("06", cac2$Maio1)-1):c(grep("07", cac2$Maio1)-2),c(1,5,9,13)] <-  
"06" #encontra dia 06 e preenche com 06 da celula anterior ate duas celulas  
antes do dia 07 (esse eh o intervalo correspondente ao dia no dataframe)
```

```
cac2[c(grep("07", cac2$Maio1)-1):c(grep("08", cac2$Maio1)-2),c(1,5,9,13)] <-  
"07" #encontra dia 07 e preenche com 07 da celula anterior ate duas celulas  
antes do dia 08 (esse eh o intervalo correspondente ao dia no dataframe)
```



```
cac2[c(grep("08", cac2$Maio1)-1):c(grep("09", cac2$Maio1)-2),c(1,5,9,13)] <-  
"08" #encontra dia 08 e preenche com 08 da celula anterior ate duas celulas  
antes do dia 09 (esse eh o intervalo correspondente ao dia no dataframe)  
  
cac2[c(grep("09", cac2$Maio1)-1):c(grep("10", cac2$Maio1)-2),c(1,5,9,13)] <-  
"09" #encontra dia 09 e preenche com 09 da celula anterior ate duas celulas  
antes do dia 10 (esse eh o intervalo correspondente ao dia no dataframe)  
  
cac2[c(grep("10", cac2$Maio1)-1):c(grep("11", cac2$Maio1)-2),c(1,5,9,13)] <-  
"10" #encontra dia 10 e preenche com 10 da celula anterior ate duas celulas  
antes do dia 11 (esse eh o intervalo correspondente ao dia no dataframe)  
  
cac2[c(grep("11", cac2$Maio1)-1):c(grep("12", cac2$Maio1)-2),c(1,5,9,13)] <-  
"11" #encontra dia 11 e preenche com 11 da celula anterior ate duas celulas  
antes do dia 12 (esse eh o intervalo correspondente ao dia no dataframe)  
  
cac2[c(grep("12", cac2$Maio1)-1):c(grep("13", cac2$Maio1)-2),c(1,5,9,13)] <-  
"12" #encontra dia 12 e preenche com 12 da celula anterior ate duas celulas  
antes do dia 13 (esse eh o intervalo correspondente ao dia no dataframe)  
  
cac2[c(grep("13", cac2$Maio1)-1):c(grep("14", cac2$Maio1)-2),c(1,5,9,13)] <-  
"13" #encontra dia 13 e preenche com 13 da celula anterior ate duas celulas  
antes do dia 14 (esse eh o intervalo correspondente ao dia no dataframe)  
  
cac2[c(grep("14", cac2$Maio1)-1):c(grep("15", cac2$Maio1)-2),c(1,5,9,13)] <-  
"14" #encontra dia 14 e preenche com 14 da celula anterior ate duas celulas  
antes do dia 15 (esse eh o intervalo correspondente ao dia no dataframe)  
  
cac2[c(grep("15", cac2$Maio1)-1):c(grep("16", cac2$Maio1)-2),c(1,5,9,13)] <-  
"15" #encontra dia 15 e preenche com 15 da celula anterior ate duas celulas  
antes do dia 16 (esse eh o intervalo correspondente ao dia no dataframe)  
  
cac2[c(grep("16", cac2$Maio1)-1):nrow(cac2),1] <- "16" #encontra dia 16 e  
preenche com 16 da celula anterior ate o final das linhas do objeto (esse eh  
o intervalo correspondente ao dia no dataframe)  
  
cac2[c(grep("17", cac2$Maio2)-1):c(grep("18", cac2$Maio2)-2),c(3,7,11,15)]  
<- "17" #encontra dia 17 e preenche com 17 da celula anterior ate duas  
celulas antes do dia 18 (esse eh o intervalo correspondente ao dia no  
dataframe)  
  
cac2[c(grep("18", cac2$Maio2)-1):c(grep("19", cac2$Maio2)-2),c(3,7,11,15)]  
<- "18" #encontra dia 18 e preenche com 18 da celula anterior ate duas  
celulas antes do dia 19 (esse eh o intervalo correspondente ao dia no  
dataframe)  
  
cac2[c(grep("19", cac2$Maio2)-1):c(grep("20", cac2$Maio2)-2),c(3,7,11,15)]  
<- "19" #encontra dia 19 e preenche com 19 da celula anterior ate duas  
celulas antes do dia 20 (esse eh o intervalo correspondente ao dia no
```

dataframe)

```
cac2[c(grep("20", cac2$Maio2)-1):c(grep("21", cac2$Maio2)-2),c(3,7,11,15)]  
<- "20" #encontra dia 20 e preenche com 20 da celula anterior ate duas  
celulas antes do dia 21 (esse eh o intervalo correspondente ao dia no  
dataframe)
```

```
cac2[c(grep("21", cac2$Maio2)-1):c(grep("22", cac2$Maio2)-2),c(3,7,11,15)]  
<- "21" #encontra dia 21 e preenche com 21 da celula anterior ate duas  
celulas antes do dia 22 (esse eh o intervalo correspondente ao dia no  
dataframe)
```

```
cac2[c(grep("22", cac2$Maio2)-1):c(grep("23", cac2$Maio2)-2),c(3,7,11,15)]  
<- "22" #encontra dia 22 e preenche com 22 da celula anterior ate duas  
celulas antes do dia 23 (esse eh o intervalo correspondente ao dia no  
dataframe)
```

```
cac2[c(grep("23", cac2$Maio2)-1):c(grep("24", cac2$Maio2)-2),c(3,7,11,15)]  
<- "23" #encontra dia 23 e preenche com 23 da celula anterior ate duas  
celulas antes do dia 24 (esse eh o intervalo correspondente ao dia no  
dataframe)
```

```
cac2[c(grep("24", cac2$Maio2)-1):c(grep("25", cac2$Maio2)-2),c(3,7,11,15)]  
<- "24" #encontra dia 24 e preenche com 24 da celula anterior ate duas  
celulas antes do dia 25 (esse eh o intervalo correspondente ao dia no  
dataframe)
```

```
cac2[c(grep("25", cac2$Maio2)-1):c(grep("26", cac2$Maio2)-2),c(3,7,11,15)]  
<- "25" #encontra dia 25 e preenche com 25 da celula anterior ate duas  
celulas antes do dia 26 (esse eh o intervalo correspondente ao dia no  
dataframe)
```

```
cac2[c(grep("26", cac2$Maio2)-1):c(grep("27", cac2$Maio2)-2),c(3,7,11,15)]  
<- "26" #encontra dia 26 e preenche com 26 da celula anterior ate duas  
celulas antes do dia 27 (esse eh o intervalo correspondente ao dia no  
dataframe)
```

```
cac2[c(grep("27", cac2$Maio2)-1):c(grep("28", cac2$Maio2)-2),c(3,7,11,15)]  
<- "27" #encontra dia 27 e preenche com 27 da celula anterior ate duas  
celulas antes do dia 28 (esse eh o intervalo correspondente ao dia no  
dataframe)
```

```
cac2[c(grep("28", cac2$Maio2)-1):c(grep("29", cac2$Maio2)-2),c(3,7,11,15)]  
<- "28" #encontra dia 28 e preenche com 28 da celula anterior ate duas  
celulas antes do dia 29 (esse eh o intervalo correspondente ao dia no  
dataframe)
```

```
cac2[c(grep("29", cac2$Maio2)-1):c(grep("30", cac2$Maio2)-2),c(3,7,11,15)]  
<- "29" #encontra dia 29 e preenche com 29 da celula anterior ate duas  
celulas antes do dia 30 (esse eh o intervalo correspondente ao dia no
```

```
dataframe)
```

```
cac2[c(grep("30", cac2$Maio2)-1):c(grep("31", cac2$Maio2)-2),c(3,7,11,15)]  
<- "30" #encontra dia 30 e preenche com 30 da celula anterior ate duas  
celulas antes do dia 31 (esse eh o intervalo correspondente ao dia no  
dataframe)
```

```
cac2[c(grep("31", cac2$Maio2)-1):nrow(cac2),c(3,7,11,15)] <- "31" #encontra  
dia 31 e preenche com 31 da celula anterior ate o final das linhas do objeto  
(esse eh o intervalo correspondente ao dia no dataframe)
```

```
cac3[c(grep("01", cac3$Outubro1)-1):c(grep("02",  
cac3$Outubro1)-2),c(1,5,9,13)] <- "01" #encontra dia 01 e preenche com 01 da  
celula anterior ate duas celulas antes do dia 02 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("02", cac3$Outubro1)-1):c(grep("03",  
cac3$Outubro1)-2),c(1,5,9,13)] <- "02" #encontra dia 02 e preenche com 02 da  
celula anterior ate duas celulas antes do dia 03 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("03", cac3$Outubro1)-1):c(grep("04",  
cac3$Outubro1)-2),c(1,5,9,13)] <- "03" #encontra dia 03 e preenche com 03 da  
celula anterior ate duas celulas antes do dia 04 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("04", cac3$Outubro1)-1):c(grep("05",  
cac3$Outubro1)-2),c(1,5,9,13)] <- "04" #encontra dia 04 e preenche com 04 da  
celula anterior ate duas celulas antes do dia 05 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("05", cac3$Outubro1)-1):c(grep("06",  
cac3$Outubro1)-2),c(1,5,9,13)] <- "05" #encontra dia 05 e preenche com 05 da  
celula anterior ate duas celulas antes do dia 06 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("06", cac3$Outubro1)-1):c(grep("07",  
cac3$Outubro1)-2),c(1,5,9,13)] <- "06" #encontra dia 06 e preenche com 06 da  
celula anterior ate duas celulas antes do dia 07 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("07", cac3$Outubro1)-1):c(grep("08",  
cac3$Outubro1)-2),c(1,5,9,13)] <- "07" #encontra dia 07 e preenche com 07 da  
celula anterior ate duas celulas antes do dia 08 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("08", cac3$Outubro1)-1):c(grep("09",
```

```
cac3$0utubro1)-2),c(1,5,9,13)] <- "08" #encontra dia 08 e preenche com 08 da  
celula anterior ate duas celulas antes do dia 09 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("09", cac3$0utubro1)-1):c(grep("10",  
cac3$0utubro1)-2),c(1,5,9,13)] <- "09" #encontra dia 09 e preenche com 09 da  
celula anterior ate duas celulas antes do dia 10 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("10", cac3$0utubro1)-1):c(grep("11",  
cac3$0utubro1)-2),c(1,5,9,13)] <- "10" #encontra dia 10 e preenche com 10 da  
celula anterior ate duas celulas antes do dia 11 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("11", cac3$0utubro1)-1):c(grep("12",  
cac3$0utubro1)-2),c(1,5,9,13)] <- "11" #encontra dia 11 e preenche com 11 da  
celula anterior ate duas celulas antes do dia 12 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("12", cac3$0utubro1)-1):c(grep("13",  
cac3$0utubro1)-2),c(1,5,9,13)] <- "12" #encontra dia 12 e preenche com 12 da  
celula anterior ate duas celulas antes do dia 13 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("13", cac3$0utubro1)-1):c(grep("14",  
cac3$0utubro1)-2),c(1,5,9,13)] <- "13" #encontra dia 13 e preenche com 13 da  
celula anterior ate duas celulas antes do dia 14 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("14", cac3$0utubro1)-1):c(grep("15",  
cac3$0utubro1)-2),c(1,5,9,13)] <- "14" #encontra dia 14 e preenche com 14 da  
celula anterior ate duas celulas antes do dia 15 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("15", cac3$0utubro1)-1):c(grep("16",  
cac3$0utubro1)-2),c(1,5,9,13)] <- "15" #encontra dia 15 e preenche com 15 da  
celula anterior ate duas celulas antes do dia 16 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("16", cac3$0utubro1)-1):nrow(cac3),1] <- "16" #encontra dia 16 e  
preenche com 16 da celula anterior ate o final das linhas do objeto (esse eh  
o intervalo correspondente ao dia no dataframe)
```

```
cac3[c(grep("17", cac3$0utubro2)-1):c(grep("18",  
cac3$0utubro2)-2),c(3,7,11,15)] <- "17" #encontra dia 17 e preenche com 17  
da celula anterior ate duas celulas antes do dia 18 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("18", cac3$Outubro2)-1):c(grep("19",  
cac3$Outubro2)-2),c(3,7,11,15)] <- "18" #encontra dia 18 e preenche com 18  
da celula anterior ate duas celulas antes do dia 19 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("19", cac3$Outubro2)-1):c(grep("20",  
cac3$Outubro2)-2),c(3,7,11,15)] <- "19" #encontra dia 19 e preenche com 19  
da celula anterior ate duas celulas antes do dia 20 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("20", cac3$Outubro2)-1):c(grep("21",  
cac3$Outubro2)-2),c(3,7,11,15)] <- "20" #encontra dia 20 e preenche com 20  
da celula anterior ate duas celulas antes do dia 21 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("21", cac3$Outubro2)-1):c(grep("22",  
cac3$Outubro2)-2),c(3,7,11,15)] <- "21" #encontra dia 21 e preenche com 21  
da celula anterior ate duas celulas antes do dia 22 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("22", cac3$Outubro2)-1):c(grep("23",  
cac3$Outubro2)-2),c(3,7,11,15)] <- "22" #encontra dia 22 e preenche com 22  
da celula anterior ate duas celulas antes do dia 23 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("23", cac3$Outubro2)-1):c(grep("24",  
cac3$Outubro2)-2),c(3,7,11,15)] <- "23" #encontra dia 23 e preenche com 23  
da celula anterior ate duas celulas antes do dia 24 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("24", cac3$Outubro2)-1):c(grep("25",  
cac3$Outubro2)-2),c(3,7,11,15)] <- "24" #encontra dia 24 e preenche com 24  
da celula anterior ate duas celulas antes do dia 25 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("25", cac3$Outubro2)-1):c(grep("26",  
cac3$Outubro2)-2),c(3,7,11,15)] <- "25" #encontra dia 25 e preenche com 25  
da celula anterior ate duas celulas antes do dia 26 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("26", cac3$Outubro2)-1):c(grep("27",  
cac3$Outubro2)-2),c(3,7,11,15)] <- "26" #encontra dia 26 e preenche com 26  
da celula anterior ate duas celulas antes do dia 27 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("27", cac3$Outubro2)-1):c(grep("28",  
cac3$Outubro2)-2),c(3,7,11,15)] <- "27" #encontra dia 27 e preenche com 27  
da celula anterior ate duas celulas antes do dia 28 (esse eh o intervalo  
correspondente ao dia no dataframe)
```

```
cac3[c(grep("28", cac3$Outubro2)-1):c(grep("29",
```

```
cac3$Outubro2)-2),c(3,7,11,15)] <- "28" #encontra dia 28 e preenche com 28
da celula anterior ate duas celulas antes do dia 29 (esse eh o intervalo
correspondente ao dia no dataframe)

cac3[c(grep("29", cac3$Outubro2)-1):c(grep("30",
cac3$Outubro2)-2),c(3,7,11,15)] <- "29" #encontra dia 29 e preenche com 29
da celula anterior ate duas celulas antes do dia 30 (esse eh o intervalo
correspondente ao dia no dataframe)

cac3[c(grep("30", cac3$Outubro2)-1):c(grep("31",
cac3$Outubro2)-2),c(3,7,11,15)] <- "30" #encontra dia 30 e preenche com 30
da celula anterior ate duas celulas antes do dia 31 (esse eh o intervalo
correspondente ao dia no dataframe)

cac3[c(grep("31", cac3$Outubro2)-1):nrow(cac3),c(3,7,11,15)] <- "31"
#encontra dia 31 e preenche com 31 da celula anterior ate o final das linhas
do objeto (esse eh o intervalo correspondente ao dia no dataframe)
```

# 4. Fazendo o subset nos dataframes de acordo com o periodo e mare selecionada

```
if (periodo == "sem.1") #condiciona caso o periodo selecionado seja
semestre 1

{

    datas <- cac[grep(mare,cac$Janeiro3, fixed = T),] #busca a
altura de mare especificada nos primeiros 16 dias de janeiro e guarda no
objeto datas

    datas <- rbind(datas, cac[grep(mare,cac$Janeiro4, fixed = T),])
#busca a altura de mare especificada nos ultimos 15 dias de janeiro e
adiciona as linhas com rbind no objeto datas

    datas <- rbind(datas, cac[grep(mare,cac$Fevereiro3, fixed =
T),]) #busca a altura de mare especificada nos primeiros 16 dias de
fevereiro e adiciona as linhas com rbind no objeto datas

    datas <- rbind(datas, cac[grep(mare,cac$Fevereiro4, fixed =
T),]) #busca a altura de mare especificada nos ultimos 12 dias de fevereiro
e adiciona as linhas com rbind no objeto datas

    datas <- rbind(datas, cac[grep(mare,cac$Março3, fixed = T),])
#busca a altura de mare especificada nos primeiros 16 dias de marco e
```

adiciona as linhas com rbind no objeto datas

```
      datas <- rbind(datas, cac[grep(mare,cac$Março4, fixed = T),])  
#busca a altura de mare especificada nos ultimos 15 dias de marco e adiciona  
as linhas com rbind no objeto datas
```

```
      datas <- rbind(datas, cac[grep(mare,cac$Abril3, fixed = T),])  
#busca a altura de mare especificada nos primeiros 16 dias de abril e  
adiciona as linhas com rbind no objeto datas
```

```
      datas <- rbind(datas, cac[grep(mare,cac$Abril4, fixed = T),])  
#busca a altura de mare especificada nos 15 dias de abril e adiciona as  
linhas com rbind no objeto datas
```

```
      datas2 <- cac2[grep(mare,cac2$Maio3, fixed = T),c(1:8)] #busca a  
altura de mare especificada nos primeiros 16 dias de maio e guarda no objeto  
datas2
```

```
      datas2 <- rbind(datas2, cac2[grep(mare,cac2$Maio4, fixed =  
T),c(1:8)]) #busca a altura de mare especificada nos ultimos 15 dias de maio  
e adiciona as linhas com rbind no objeto datas2
```

```
      datas2 <- rbind(datas2, cac2[grep(mare,cac2$Junho3, fixed =  
T),c(1:8)]) #busca a altura de mare especificada nos primeiros 16 dias de  
junho e adiciona as linhas com rbind no objeto datas2
```

```
      datas2 <- rbind(datas2, cac2[grep(mare,cac2$Junho4, fixed =  
T),c(1:8)]) #busca a altura de mare especificada nos ultimos 15 dias de  
junho e adiciona as linhas com rbind no objeto datas2
```

```
      mares.p <- c(datas, datas2) # seleciona os meses correspondentes  
ao semestre 1 guardados nos objetos datas e datas2 e guarda na lista mares.p
```

```
    }
```

```
    if (periodo == "sem.2") #condiciona caso o periodo selecionado seja  
semestre 2
```

```
    {
```

```
      datas2 <- cac2[grep(mare,cac2$Julho3, fixed = T),c(9:16)] #busca  
a altura de mare especificada nos primeiros 16 dias de julho e adiciona as  
linhas com rbind no objeto datas2
```

```
      datas2 <- rbind(datas2, cac2[grep(mare,cac2$Julho4, fixed =  
T),c(9:16)]) #busca a altura de mare especificada nos ultimos 15 dias julho  
e adiciona as linhas com rbind no objeto datas2
```

```
      datas2 <- rbind(datas2, cac2[grep(mare,cac2$Agosto3, fixed =  
T),c(9:16)]) #busca a altura de mare especificada nos primeiros 16 dias de
```

abril e adiciona as linhas com rbind no objeto datas2

```
datas2 <- rbind(datas2, cac2[grep(mare,cac2$Agosto4, fixed =  
T),c(9:16)]) #busca a altura de mare especificada nos ultimos 14 dias de  
agosto e adiciona as linhas com rbind no objeto datas2
```

```
datas3 <- cac3[grep(mare,cac3$Setembro3, fixed = T),] #busca a  
altura de mare especificada nos primeiros 16 dias de setembro e guarda no  
objeto datas3
```

```
datas3 <- rbind(datas3, cac3[grep(mare,cac3$Setembro4, fixed =  
T),]) #busca a altura de mare especificada nos ultimos 14 dias de setembro e  
adiciona as linhas com rbind no objeto datas3
```

```
datas3 <- rbind(datas3, cac3[grep(mare,cac3$Outubro3, fixed =  
T),]) #busca a altura de mare especificada nos primeiros 16 dias de outubro  
e adiciona as linhas com rbind no objeto datas3
```

```
datas3 <- rbind(datas3, cac3[grep(mare,cac3$Outubro4, fixed =  
T),]) #busca a altura de mare especificada nos ultimos 15 dias de outubro e  
adiciona as linhas com rbind no objeto datas3
```

```
datas3 <- rbind(datas3, cac3[grep(mare,cac3$Novembro3, fixed =  
T),]) #busca a altura de mare especificada nos primeiros 16 dias de novembro  
e adiciona as linhas com rbind no objeto datas3
```

```
datas3 <- rbind(datas3, cac3[grep(mare,cac3$Novembro4, fixed =  
T),]) #busca a altura de mare especificada nos ultimos 15 dias de novembro e  
adiciona as linhas com rbind no objeto datas3
```

```
datas3 <- rbind(datas3, cac3[grep(mare,cac3$Dezembro3, fixed =  
T),]) #busca a altura de mare especificada nos primeiros 16 dias de dezembro  
e adiciona as linhas com rbind no objeto datas3
```

```
datas3 <- rbind(datas3, cac3[grep(mare,cac3$Dezembro4, fixed =  
T),]) #busca a altura de mare especificada nos ultimos 15 dias de dezembro e  
adiciona as linhas com rbind no objeto datas3
```

```
mares.p <- c(datas2, datas3) # seleciona os meses  
correspondentes ao semestre 2 guardados nos objetos datas2 e datas3 e guarda  
na lista mares.p
```

```
}
```

```
if (periodo == "tri.1") #condiciona caso o periodo selecionado seja  
trimestre 1
```

```
{
```



```
      datas <- cac[grep(mare,cac$Janeiro3, fixed = T),c(1:12)] #busca
a altura de mare especificada nos primeiros 16 dias de janeiro e guarda no
objeto datas

      datas <- rbind(datas, cac[grep(mare,cac$Janeiro4, fixed =
T),c(1:12)]) #busca a altura de mare especificada nos ultimos 15 dias de
janeiro e adiciona as linhas com rbind no objeto datas

      datas <- rbind(datas, cac[grep(mare,cac$Fevereiro3, fixed =
T),c(1:12)]) #busca a altura de mare especificada nos primeiros 16 dias de
fevereiro e adiciona as linhas com rbind no objeto datas

      datas <- rbind(datas, cac[grep(mare,cac$Fevereiro4, fixed =
T),c(1:12)]) #busca a altura de mare especificada nos ultimos 12 dias de
fevereiro e adiciona as linhas com rbind no objeto datas

      datas <- rbind(datas, cac[grep(mare,cac$Março3, fixed =
T),c(1:12)]) #busca a altura de mare especificada nos primeiros 16 dias de
marco e adiciona as linhas com rbind no objeto datas

      datas <- rbind(datas, cac[grep(mare,cac$Março4, fixed =
T),c(1:12)]) #busca a altura de mare especificada nos ultimos 15 dias de
marco e adiciona as linhas com rbind no objeto datas

      mares.p <- datas # seleciona os meses correspondentes ao
trimestre 1 guardados no objeto datas e guarda na lista mares.p

    }

    if (periodo == "tri.2")

    {

      datas <- cac[grep(mare,cac$Abril3, fixed = T),c(13:16)] #busca a
altura de mare especificada nos primeiros 16 dias de abril e adiciona as
linhas com rbind no objeto datas

      datas <- rbind(datas, cac[grep(mare,cac$Abril4, fixed =
T),c(13:16)]) #busca a altura de mare especificada nos 15 dias de abril e
adiciona as linhas com rbind no objeto datas

      datas2 <- cac2[grep(mare,cac2$Maio3, fixed = T),c(1:8)] #busca a
altura de mare especificada nos primeiros 16 dias de maio e guarda no objeto
datas2

      datas2 <- rbind(datas2, cac2[grep(mare,cac2$Maio4, fixed =
T),c(1:8)]) #busca a altura de mare especificada nos ultimos 15 dias de maio
e adiciona as linhas com rbind no objeto datas2

      datas2 <- rbind(datas2, cac2[grep(mare,cac2$Junho3, fixed =
```

```
T),c(1:8)]) #busca a altura de mare especificada nos primeiros 16 dias de
junho e adiciona as linhas com rbind no objeto datas2

      datas2 <- rbind(datas2, cac2[grep(mare,cac2$Junho4, fixed =
T),c(1:8)]) #busca a altura de mare especificada nos ultimos 15 dias de
junho e adiciona as linhas com rbind no objeto datas2

      mares.p <- c(datas, datas2) # seleciona os meses correspondentes
ao trimestre 2 guardados nos objetos datas e datas2 e guarda na lista
mares.p

    }

    if (periodo == "tri.3")

    {

      datas2 <- cac2[grep(mare,cac2$Julho3, fixed = T),c(9:16)] #busca
a altura de mare especificada nos primeiros 16 dias de julho e adiciona as
linhas com rbind no objeto datas2

      datas2 <- rbind(datas2, cac2[grep(mare,cac2$Julho4, fixed =
T),c(9:16)]) #busca a altura de mare especificada nos ultimos 15 dias julho
e adiciona as linhas com rbind no objeto datas2

      datas2 <- rbind(datas2, cac2[grep(mare,cac2$Agosto3, fixed =
T),c(9:16)]) #busca a altura de mare especificada nos primeiros 16 dias de
abril e adiciona as linhas com rbind no objeto datas2

      datas2 <- rbind(datas2, cac2[grep(mare,cac2$Agosto4, fixed =
T),c(9:16)]) #busca a altura de mare especificada nos ultimos 14 dias de
agosto e adiciona as linhas com rbind no objeto datas2

      datas3 <- cac3[grep(mare,cac3$Setembro3, fixed = T),c(1:4)]
#busca a altura de mare especificada nos primeiros 16 dias de setembro e
guarda no objeto datas3

      datas3 <- rbind(datas3, cac3[grep(mare,cac3$Setembro4, fixed =
T),c(1:4)]) #busca a altura de mare especificada nos ultimos 14 dias de
setembro e adiciona as linhas com rbind no objeto datas3

      mares.p <- c(datas2, datas3) #seleciona os meses
correspondentes ao trimestre 3 guardados nos objetos datas2 e datas3 e
guarda na lista mares.p

    }

    if (periodo == "tri.4")
```

```
{

    datas3 <- cac3[grep(mare,cac3$Outubro3, fixed = T),c(5:16)]
#busca a altura de mare especificada nos primeiros 16 dias de outubro e
adiciona as linhas com rbind no objeto datas3

    datas3 <- rbind(datas3, cac3[grep(mare,cac3$Outubro4, fixed =
T),c(5:16)]) #busca a altura de mare especificada nos ultimos 15 dias de
outubro e adiciona as linhas com rbind no objeto datas3

    datas3 <- rbind(datas3, cac3[grep(mare,cac3$Novembro3, fixed =
T),c(5:16)]) #busca a altura de mare especificada nos primeiros 16 dias de
novembro e adiciona as linhas com rbind no objeto datas3

    datas3 <- rbind(datas3, cac3[grep(mare,cac3$Novembro4, fixed =
T),c(5:16)]) #busca a altura de mare especificada nos ultimos 15 dias de
novembro e adiciona as linhas com rbind no objeto datas3

    datas3 <- rbind(datas3, cac3[grep(mare,cac3$Dezembro3, fixed =
T),c(5:16)]) #busca a altura de mare especificada nos primeiros 16 dias de
dezembro e adiciona as linhas com rbind no objeto datas3

    datas3 <- rbind(datas3, cac3[grep(mare,cac3$Dezembro4, fixed =
T),c(5:16)]) #busca a altura de mare especificada nos ultimos 15 dias de
dezembro e adiciona as linhas com rbind no objeto datas3

    mares.p <- datas3 #seleciona os meses correspondentes ao
trimestre 4 guardados no objeto datas3 e guarda na lista mares.p

}

if (periodo == "jan")

{

    datas <- cac[grep(mare,cac$Janeiro3, fixed = T),c(1:4)] #busca a
altura de mare especificada nos primeiros 16 dias de janeiro e guarda no
objeto datas

    datas <- rbind(datas, cac[grep(mare,cac$Janeiro4, fixed =
T),c(1:4)]) #busca a altura de mare especificada nos ultimos 15 dias de
janeiro e adiciona as linhas com rbind no objeto datas

    mares.p <- datas #seleciona as colunas correspondentes a janeiro
guardados no objeto datas e guarda na lista mares.p

}
```

```
if (período == "fev")

{

    datas <- cac[grep(mare,cac$Fevereiro3, fixed = T),c(5:8)] #busca
a altura de mare especificada nos primeiros 16 dias de fevereiro e adiciona
as linhas com rbind no objeto datas

    datas <- rbind(datas, cac[grep(mare,cac$Fevereiro4, fixed =
T),c(5:8)]) #busca a altura de mare especificada nos últimos 12 dias de
fevereiro e adiciona as linhas com rbind no objeto datas

    mares.p <- datas #seleciona as colunas correspondentes a
fevereiro guardados no objeto datas e guarda na lista mares.p

}

if (período == "mar")

{

    datas <- rbind(datas, cac[grep(mare,cac$Março3, fixed =
T),c(9:12)]) #busca a altura de mare especificada nos primeiros 16 dias de
março e adiciona as linhas com rbind no objeto datas

    datas <- rbind(datas, cac[grep(mare,cac$Março4, fixed =
T),c(9:12)]) #busca a altura de mare especificada nos últimos 15 dias de
março e adiciona as linhas com rbind no objeto datas

    mares.p <- datas #seleciona as colunas correspondentes a março
guardados no objeto datas e guarda na lista mares.p

}

if (período == "abr")

{

    datas <- cac[grep(mare,cac$Abril3, fixed = T),c(13:16)] #busca a
altura de mare especificada nos primeiros 16 dias de abril e adiciona as
linhas com rbind no objeto datas

    datas <- rbind(datas, cac[grep(mare,cac$Abril4, fixed =
T),c(13:16)]) #busca a altura de mare especificada nos 15 dias de abril e
adiciona as linhas com rbind no objeto datas
```

```
        mares.p <- datas #seleciona as colunas correspondentes a abril
guardados no objeto datas e guarda na lista mares.p

    }

    if (periodo == "mai")

    {

        datas2 <- cac2[grep(mare,cac2$Maio3, fixed = T),c(1:4)] #busca a
altura de mare especificada nos primeiros 16 dias de maio e guarda no objeto
datas2

        datas2 <- rbind(datas2, cac2[grep(mare,cac2$Maio4, fixed =
T),c(1:4)]) #busca a altura de mare especificada nos ultimos 15 dias de maio
e adiciona as linhas com rbind no objeto datas2

        mares.p <- datas2 #seleciona as colunas correspondentes a maio
guardados no objeto datas e guarda na lista mares.p

    }

    if (periodo == "jun")

    {

        datas2 <- cac2[grep(mare,cac2$Junho3, fixed = T),c(5:8)] #busca
a altura de mare especificada nos primeiros 16 dias de junho e adiciona as
linhas com rbind no objeto datas2

        datas2 <- rbind(datas2, cac2[grep(mare,cac2$Junho4, fixed =
T),c(5:8)]) #busca a altura de mare especificada nos ultimos 15 dias de
junho e adiciona as linhas com rbind no objeto datas2

        mares.p <- datas2 #seleciona as colunas correspondentes a jun
guardados no objeto datas2 e guarda na lista mares.p

    }

    if (periodo == "jul")

    {

        datas2 <- cac2[grep(mare,cac2$Julho3, fixed = T),c(9:12)] #busca
a altura de mare especificada nos primeiros 16 dias de julho e adiciona as
linhas com rbind no objeto datas2

        datas2 <- rbind(datas2, cac2[grep(mare,cac2$Julho4, fixed =
```

```
T),c(9:12)]) #busca a altura de mare especificada nos ultimos 15 dias julho  
e adiciona as linhas com rbind no objeto datas2
```

```
        mares.p <- datas2 #seleciona as colunas correspondentes a julho  
o guardados no objeto datas2 e guarda na lista mares.p
```

```
    }
```

```
    if (periodo == "ago")
```

```
    {
```

```
        datas2 <- cac2[grep(mare,cac2$Agosto3, fixed = T),c(13:16)]  
#busca a altura de mare especificada nos primeiros 16 dias de abril e  
adiciona as linhas com rbind no objeto datas2
```

```
        datas2 <- rbind(datas2, cac2[grep(mare,cac2$Agosto4, fixed =  
T),c(13:16)]) #busca a altura de mare especificada nos ultimos 14 dias de  
agosto e adiciona as linhas com rbind no objeto datas2
```

```
        mares.p <- datas2 #seleciona as colunas correspondentes a agosto  
o guardados no objeto datas2 e guarda na lista mares.p
```

```
    }
```

```
    if (periodo == "set")
```

```
    {
```

```
        datas3 <- cac3[grep(mare,cac3$Setembro3, fixed = T),c(1:4)]  
#busca a altura de mare especificada nos primeiros 16 dias de setembro e  
guarda no objeto datas3
```

```
        datas3 <- rbind(datas3, cac3[grep(mare,cac3$Setembro4, fixed =  
T),c(1:4)]) #busca a altura de mare especificada nos ultimos 14 dias de  
setembro e adiciona as linhas com rbind no objeto datas3
```

```
        mares.p <- datas3 #seleciona as colunas correspondentes a  
setembro o guardados no objeto datas2 e guarda na lista mares.p
```

```
    }
```

```
    if (periodo == "out")
```

```
    {
```

```
        datas3 <- cac3[grep(mare,cac3$Outubro3, fixed = T),c(5:8)]
```

```
#busca a altura de mare especificada nos primeiros 16 dias de outubro e  
adiciona as linhas com rbind no objeto datas3
```

```
    datas3 <- rbind(datas3, cac3[grepl(mare,cac3$Outubro4, fixed =  
T),c(5:8)]) #busca a altura de mare especificada nos ultimos 15 dias de  
outubro e adiciona as linhas com rbind no objeto datas3
```

```
    mares.p <- datas3 #seleciona as colunas correspondentes a  
outubro o guardados no objeto datas3 e guarda na lista mares.p
```

```
  }
```

```
  if (periodo == "nov")
```

```
  {
```

```
    datas3 <- cac3[grepl(mare,cac3$Novembro3, fixed = T),c(9:12)]  
#busca a altura de mare especificada nos primeiros 16 dias de novembro e  
adiciona as linhas com rbind no objeto datas3
```

```
    datas3 <- rbind(datas3, cac3[grepl(mare,cac3$Novembro4, fixed =  
T),c(9:12)]) #busca a altura de mare especificada nos ultimos 15 dias de  
novembro e adiciona as linhas com rbind no objeto datas3
```

```
    mares.p <- datas3 #seleciona as colunas correspondentes a  
novembro o guardados no objeto datas3 e guarda na lista mares.p
```

```
  }
```

```
  if (periodo == "dez")
```

```
  {
```

```
    datas3 <- cac3[grepl(mare,cac3$Dezembro3, fixed = T),c(13:16)]  
#busca a altura de mare especificada nos primeiros 16 dias de dezembro e  
adiciona as linhas com rbind no objeto datas3
```

```
    datas3 <- rbind(datas3, cac3[grepl(mare,cac3$Dezembro4, fixed =  
T),c(13:16)]) #busca a altura de mare especificada nos ultimos 15 dias de  
dezembro e adiciona as linhas com rbind no objeto datas3
```

```
    mares.p <- datas3 #seleciona as colunas correspondentes a  
dezembro o guardados no objeto datas3 e guarda na lista mares.p
```

```
  }
```

```
r <- lapply(mares.p, `length<-`, max(lengths(mares.p))) #deixa todos os
elementos da lista com o mesmo numero de linhas para que possa ser
transformado em dataframe

r <- as.data.frame(r) #transforma a lista em dataframe para que quando seja
salvo a formatacao fique mais palatavel para o usuario

write.table(r, "resultados_mare.csv", sep= ";", row.names = F) #guarda o
dataframe em um arquivo .csv nomeado como "resultados_mare.csv", separado
por ; que eh o padrao da maioria de softwares de leitura

x <- readline("As datas com os resultados selecionados estao salvas no
documento resultados_mare.csv, no diretorio de trabalho do R\n(pressione
enter para continuar)") #imprime a tela a mensagem e aguarda o usuario
pressionar enter (assim ela fica na tela tempo suficiente para ser lida
antes de retornar o dataframe resultante)

return(r) #retorna o dataframe resultado na tela

}
```

Link para o arquivo .txt com o código da função [f\\_mare\\_final.r](#)

From:  
<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:  
[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2018:alunos:trabalho\\_final:cecilimendes:funcao\\_mares](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:cecilimendes:funcao_mares)

Last update: **2020/08/12 06:04**