---

title: "PAINEL DE GARANTIAS "

runtime: shiny

output:

flexdashboard::flex\_dashboard:

navbar:

- { title: "Data: 30/04/2019",icon: ion-ios-calendar-outline, href: "https://example.com/about" ,align: right }

orientation: columns

vertical\_layout: fill

---

```{r setup, include=FALSE}

#library(ggplot2)

#library(shiny)

library(plotly)

library(flexdashboard)

#library(knitr)

#library(png)

#library(readxl)

#library(tidyr)

#library(lubridate)

library(stringr)

library(dplyr)

#library(shinydashboard)

#library(matrixStats)

#library(rlang)

#library(rmarkdown)

library(kableExtra)

#library(hpackedbubble)

load("Garantias.Rdata", envir=.GlobalEnv)

funcao\_formata\_quantidade <- function(x){

paste0(format(round(x/1000000, 2), big.mark = ".", decimal.mark = ","))

}

funcao\_formata\_quantidade\_2 <- function(x){

paste0(format(round(x/1000000, 0), big.mark = ".", decimal.mark = ","))

}

formata <- function(x){

paste0("R$", format(round(x, 2), big.mark = ".", decimal.mark = ","), " mi")

}

funcao\_percentual\_taxa <- function(x){

paste0(format(round(x\*100, 2), big.mark = ".", decimal.mark = ","),"%")

}

numero <- function(x){

a <- as.character(x)

b <- str\_replace\_all(a,"\\.","")

c <- str\_replace\_all(b,",","\\.")

d <- as.numeric(c)

return(d)

}

```

Estados

=======================================================================

Column

-------------------------------------

<style>

.card {

background-color: #f2f2f2;

color: black;

font-family: monospace;

font-size: 12pt;

font-wight: bold;

line-height: 15pt;

margin-botton: 5px;

border-radius: 2px;

box-shadow: 10px;

}

</style>

<style>

.header {

background-color: #f5f5f0;

color: black;

font-family: sans-serif;

font-size: 12pt;

font-wight: bold;

line-height: 15pt;

margin-botton: 15px;

border-radius: 2px;

box-shadow: 10px;

float: left;

width: 100%;

height: 1%;

line-height:30pt;

box-sizing: border-box;

}

</style>

### {.card}

<style>

#seletor {

background-color: #cccc00;

}

</style>

<style>

label.control-label,

selectize-control.single {

display: table-cell;

text-align: center;

vertical-align: middle;

}

label.control-label {

padding-right: 1px;

}

.form-group {

display: table-row;

}

.selectize-control.single div.item {

padding-right: 8px;

}

.selectize-input {

height: 40px;

width: 200px;

padding-top: 5px;

}

</style>

```{r grafico, warning=FALSE, message=FALSE }

#ESTADOS - BASE DE DADOS: formatar e filtrar

grupos <- c("Todas","Garantia Total","Estados", "Bancos Federais", "Municípios","Estatais Federais","Entidades Estaduais Controladas","Empresas Privatizadas")

agrupador\_total\_formatado <- agrupador\_total %>%

filter(!(Inicio %in% grupos)) %>%

mutate(total\_total = as.character(total\_total),

total\_total = str\_replace\_all(total\_total,"\\.",""),

total\_total = str\_replace\_all(total\_total,",","\\."),

total\_total = as.numeric(total\_total),

interna\_total = numero(interna\_total),

interna\_USD = numero(interna\_USD),

externa\_total = numero(externa\_total)

)

agrupador\_formatado\_estados <- agrupador\_total %>%

filter(!(Inicio %in% grupos)) %>%

filter((Classificador == "Estados")) %>%

mutate(total\_total = numero(total\_total), interna\_total = numero(interna\_total), interna\_USD = numero(interna\_USD), externa\_total = numero(externa\_total), total\_cambial = numero(total\_cambial)) %>%

mutate(media\_total\_estados = mean(total\_total))

teste <- agrupador\_formatado\_estados[order(-rank(agrupador\_formatado\_estados$total\_total)),]

agrupador\_ATM\_estados\_formatado <- agrupador\_atm\_completo %>%

filter(!(Inicio %in% grupos)) %>%

filter((Classificador == "Estados")) %>%

mutate(ATM\_Total = numero(ATM\_Total))

agrupador\_custo\_estados\_formatado <- agrupador\_custo\_completo %>%

filter(!(Inicio %in% grupos)) %>%

filter((Classificador == "Estados")) %>%

mutate(Custo\_total\_formatado = numero(str\_replace\_all(Custo\_Total,"%","")))

```

```{r Botoes, warning=FALSE, message=FALSE }

fluidPage(

fluidRow(

column(5,div(selectInput("Mutuario",

label="",

choices = agrupador\_formatado\_estados$Inicio), id='seletor'))

#column(width=2,align="center",div(style = "height:40px;font-weight: bold;background-color: ;","CAPAG")),

#column(width=1,align="center",div(style = "height:40px;background-color: ;",textOutput("Capag")))

))

botoes <- reactive({

list(input$Mutuario)

})

observeEvent(botoes(),{

if (is.null(input$Mutuario)){entrada <- agrupador\_formatado\_estados$Inicio[1]}

else

{entrada <- input$Mutuario}

if(entrada != "all"){

filtro\_estados <- agrupador\_formatado\_estados %>%

filter(Inicio == entrada) }

})

```

```{r card\_estados, warning=FALSE, message=FALSE }

fluidPage(

fluidRow(

br(),

column(width=2,align="left",div(style = "height:40px;font-size: 13pt;font-weight: bold;background-color: ;","CAPAG")),

column(width=1,align="center",div(style = "height:40px;font-size: 13pt;font-weight: bold;background-color: ;",textOutput("Capag"))),

br(), br(),

column(width=6,align="center",div(style = "height:17px;font-size: 12pt;background-color: ;","")),

column(width=3,align="left",div(style = "height:17px;font-size: 12pt;background-color: ;","R$ Milhões")),

br(),

column(width=6,align="left",div(style = "height:17px;font-size: 13pt;font-weight: bold;background-color: ;","OPERAÇÕES GARANTIDAS")),

column(width=2,align="right",div(style = "height:17px;font-size: 13pt;font-weight: bold;background-color: ;",textOutput("Garantia\_Total\_2"))),

#column(width=2,align="right",div(style = "height:17px;background-color: ;","(R$ Milhões)")),

#column(width=3,align="center",div(style = "height:40px;background-color: ;","CAPAG")),

#column(width=3,align="center",div(style = "height:40px;background-color: ;",textOutput("Capag"))),

br(),

column(width=6,align="left",div(style = "height:17px;font-size: 13pt;background-color: ;","INTERNAS")),

column(width=2,align="right",div(style = "height:17px;font-size: 13pt;background-color: ;",textOutput("Garantia\_interna\_2"))),

#column(width=2,align="center",div(style = "height:17px;background-color: ;","EXTERNAS")),

# column(width=4,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_externa\_2"))),

br(),

column(width=1,align="left",div(style = "height:17px;font-size: 12pt;background-color: ;","")),

column(width=5,align="center",div(style = "height:17px;color: #737373;background-color: ;","Internas Cambiais:")),

column(width=2,align="right",div(style = "height:17px;color: #737373;font-size: 13pt;background-color: ;",textOutput("Garantia\_interna\_cambial\_2"))),

br(),

column(width=1,align="right",div(style = "height:17px;font-size: 12pt;background-color: ;","")),

column(width=4,align="right",div(style = "height:17px;color: #737373;background-color: ;","Internas Demais:")),

column(width=1,align="right",div(style = "height:17px;font-size: 12pt;background-color: ;","")),

column(width=2,align="right",div(style = "height:17px;color: #737373;font-size: 13pt;background-color: ;",textOutput("Garantia\_interna\_demais\_2"))),

br(),

column(width=6,align="left",div(style = "height:17px;font-size: 13pt;background-color: ;","EXTERNAS")),

column(width=2,align="right",div(style = "height:17px;font-size: 13pt;background-color: ;",textOutput("Garantia\_externa\_2"))),

br(), br(),

column(width=5,align="left",div(style = "height:17px;font-size: 13pt;font-weight: bold;background-color: ;","VIDA MÉDIA - ATM")),

column(width=3,align="right",div(style = "height:17px;font-size: 13pt;font-weight: bold;background-color: ;",textOutput("Garantia\_ATM"))),

br(),br(),

column(width=5,align="left",div(style = "height:17px;font-size: 13pt;font-weight: bold;background-color: ;","CUSTO MÉDIO")),

column(width=3,align="right",div(style = "height:17px;font-size: 13pt;font-weight: bold;background-color: ;",textOutput("Garantia\_custo"))),

br(),br(),

column(width=6,align="left",div(style = "height:17px;font-size: 13pt;font-weight: bold;background-color: ;","PERCENTUAL VINCENDO")),

column(width=2,align="center",div(style = "height:17px;font-size: 12pt;background-color: ;","")),

column(width=3,align="right",div(style = "height:17px;font-size: 12pt;background-color: ;","R$ Milhões")),

br(),

column(width=4,align="left",div(style = "height:17px;background-color: ;","Até 12 meses")),

column(width=4,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_12\_meses\_percentual"))),

column(width=3,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_12\_meses"))),

br(),

column(width=4,align="left",div(style = "height:17px;background-color: ;","De 1 a 2 anos")),

column(width=4,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_1\_ano\_percentual"))),

column(width=3,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_1\_ano"))),

br(),

column(width=4,align="left",div(style = "height:17px;background-color: ;","De 2 a 3 anos")),

column(width=4,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_2\_ano\_percentual"))),

column(width=3,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_2\_ano"))),

br(),

column(width=4,align="left",div(style = "height:17px;background-color: ;","De 3 a 4 anos")),

column(width=4,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_3\_ano\_percentual"))),

column(width=3,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_3\_ano"))),

br(),

column(width=4,align="left",div(style = "height:17px;background-color: ;","De 4 a 5 anos")),

column(width=4,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_4\_ano\_percentual"))),

column(width=3,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_4\_ano"))),

br(),

column(width=4,align="left",div(style = "height:17px;background-color: ;","Mais de 5 anos")),

column(width=4,align="right",div(style = "height:17px;;background-color: ;",textOutput("Garantia\_5\_ano\_percentual"))),

column(width=3,align="right",div(style = "height:17px;;background-color: ;",textOutput("Garantia\_5\_ano")))

#column(width=4,align="left",div(style = "height:17px;background-color: ;","CONTRATOS ASSINADOS"))

))

Garantia\_Total\_reativo <- reactive({

list(input$Mutuario)

})

Garantia\_ATM\_reativo <- reactive({

list(input$Mutuario)

})

Garantia\_Custo\_reativo <- reactive({

list(input$Mutuario)

})

Garantia\_Vincendo\_reativo <- reactive({

list(input$Mutuario)

})

observeEvent(Garantia\_Total\_reativo(),{

if (is.null(input$Mutuario)){entrada <- "agrupador\_total$Classificador[1]"}

else

{entrada <- input$Mutuario}

if(entrada != "all"){

filtro\_classificador\_estado <- agrupador\_formatado\_estados %>%

filter(Inicio == entrada) }

output$Garantia\_Total\_2 <- renderText(paste0(funcao\_formata\_quantidade(filtro\_classificador\_estado$total\_total)))

output$Garantia\_interna\_2 <- renderText(paste0(funcao\_formata\_quantidade(filtro\_classificador\_estado$interna\_total)))

output$Garantia\_interna\_cambial\_2 <- renderText(paste0(funcao\_formata\_quantidade(filtro\_classificador\_estado$interna\_USD)))

output$Garantia\_interna\_demais\_2 <- renderText(paste0(funcao\_formata\_quantidade(filtro\_classificador\_estado$interna\_total-filtro\_classificador\_estado$interna\_USD)))

output$Garantia\_externa\_2 <- renderText(paste0(funcao\_formata\_quantidade(filtro\_classificador\_estado$externa\_total)))

})

#ATM

observeEvent(Garantia\_ATM\_reativo(),{

if (is.null(input$Mutuario)){entrada <- "agrupador\_atm\_completo$Classificador[1]"}

else

{entrada <- input$Mutuario}

if(entrada != "all"){

filtro\_classificador\_estado\_atm <- agrupador\_ATM\_estados\_formatado %>%

filter(Inicio == entrada) }

output$Garantia\_ATM <- renderText(paste0(str\_replace\_all(round(filtro\_classificador\_estado\_atm$ATM\_Total,2),"\\.",","), " anos"))

})

#CUSTO

observeEvent(Garantia\_Custo\_reativo(),{

if (is.null(input$Mutuario)){entrada <- "agrupador\_custo\_completo$Classificador[1]"}

else

{entrada <- input$Mutuario}

if(entrada != "all"){

filtro\_classificador\_estado\_custo <- agrupador\_custo\_estados\_formatado %>%

filter(Inicio == entrada) }

output$Garantia\_custo <- renderText(paste0(filtro\_classificador\_estado\_custo$Custo\_Total, " a.a"))

})

#PERCENTUAL VINCENDO

observeEvent(Garantia\_Vincendo\_reativo(),{

if (is.null(input$Mutuario)){entrada <- "agrupador\_custo\_completo$Classificador[1]"}

else

{entrada <- input$Mutuario}

if(entrada != "all"){

filtro\_classificador\_estado\_vincendo <- agrupador\_percentual\_vincendo %>%

filter(Classificador == "Estados") %>%

filter(Inicio == entrada) }

output$Garantia\_12\_meses <- renderText(paste0(funcao\_formata\_quantidade(numero(filtro\_classificador\_estado\_vincendo$Ate\_12\_meses))))

output$Garantia\_12\_meses\_percentual <- renderText(paste0(filtro\_classificador\_estado\_vincendo$Ate\_12\_meses\_percentual))

output$Garantia\_1\_ano <- renderText(paste0(funcao\_formata\_quantidade(numero(filtro\_classificador\_estado\_vincendo$De\_1\_2\_anos))))

output$Garantia\_1\_ano\_percentual <- renderText(paste0(filtro\_classificador\_estado\_vincendo$De\_1\_2\_anos\_percentual))

output$Garantia\_2\_ano <- renderText(paste0(funcao\_formata\_quantidade(numero(filtro\_classificador\_estado\_vincendo$De\_2\_3\_anos))))

output$Garantia\_2\_ano\_percentual <- renderText(paste0(filtro\_classificador\_estado\_vincendo$De\_2\_3\_anos\_percentual))

output$Garantia\_3\_ano <- renderText(paste0(funcao\_formata\_quantidade(numero(filtro\_classificador\_estado\_vincendo$De\_3\_4\_anos))))

output$Garantia\_3\_ano\_percentual <- renderText(paste0(filtro\_classificador\_estado\_vincendo$De\_3\_4\_anos\_percentual))

output$Garantia\_4\_ano <- renderText(paste0(funcao\_formata\_quantidade(numero(filtro\_classificador\_estado\_vincendo$De\_4\_5\_anos))))

output$Garantia\_4\_ano\_percentual <- renderText(paste0(filtro\_classificador\_estado\_vincendo$De\_4\_5\_anos\_percentual))

output$Garantia\_5\_ano <- renderText(paste0(funcao\_formata\_quantidade(numero(filtro\_classificador\_estado\_vincendo$Acima\_5\_anos))))

output$Garantia\_5\_ano\_percentual <- renderText(paste0(filtro\_classificador\_estado\_vincendo$Acima\_5\_anos\_percentual))

})

```

```{r capag, warning=FALSE, message=FALSE}

#CAPAG

Garantia\_Capag\_reativo <- reactive({

list(input$Mutuario)

})

observeEvent(Garantia\_Capag\_reativo(),{

if (is.null(input$Mutuario)){entrada <- "agrupador\_custo\_completo$Classificador[1]"}

else

{entrada <- input$Mutuario}

if(entrada != "all"){

filtro\_classificador\_estado\_capag <- novos\_contratos %>%

filter(`Tipo Mutuário` == "Estados") %>%

filter(`Mutuário` == entrada) }

output$Capag <- renderText(paste0(filtro\_classificador\_estado\_capag$CAPAG[1]))

})

```

### Contratos Assinados no \*\*ano\*\* {data-height=130}

```{r Tabela, echo = FALSE, results = 'asis'}

htmlOutput("Tabela\_Estado")

Tabela\_reativo <- reactive({

list(input$Mutuario)

})

observeEvent(Tabela\_reativo(),{

if (is.null(input$Mutuario)){entrada <- "agrupador\_custo\_completo$Classificador[1]"}

else

{entrada <- input$Mutuario}

if(entrada != "all"){

filtro\_classificador\_estado\_tabela <- novos\_contratos %>%

filter(`Tipo Mutuário` == "Estados") %>%

mutate(`Valor` = paste0("R$ ",funcao\_formata\_quantidade(numero(`Valor Contratado Original`))," Milhões" )) %>%

filter(`Mutuário` == entrada) }

output$Tabela\_Estado <- renderText({

kable(filtro\_classificador\_estado\_tabela[c(17,3,9,10,26)], align = 'c') %>%

kable\_styling() %>%

scroll\_box(width = "100%", height = "135px")

})

})

```

Column {.tabset}

-----------------------------------------------------------------------

### Operações Garantidas

```{r divida total estados, warning=FALSE, message=FALSE ,fig.width=9, fig.height=7 }

plotlyOutput("divida\_total\_estados")

cor\_reativo\_divida\_total\_estados <- reactive({

list(input$Mutuario)

})

cores\_divida\_total\_estados<-reactive({ifelse(agrupador\_formatado\_estados[,1]==cor\_reativo\_divida\_total\_estados(),"#cccc00","#b3b3b3")})

output$divida\_total\_estados <-renderPlotly(

plot\_ly(agrupador\_formatado\_estados, x=~Inicio) %>%

add\_trace(y = ~media\_total\_estados/1000000 ,

type = 'scatter',

mode = 'lines',

line = list(color = '#8c8c8c', width = 1),

showlegend =TRUE,

visible = "legendonly",

name ="Média",

text = ~paste0("<b>Média:</b> ","R$ ",funcao\_formata\_quantidade(mean(total\_total))," Mihlões"),

hoverinfo = "text"

) %>%

add\_trace(y = ~total\_total/1000000,

type = 'bar',

#size=~log10(ATM\_Total),

#color = ~ATM\_Total,

#colors = c("#ffcccc","#ff6666",'#ff0000'),

name="vida",

showlegend =FALSE,

marker = list (color = cores\_divida\_total\_estados()),

#sizemode = 'diameter' #line = list(color = '#b3b3b3'

text = ~paste0("<b>Estado:</b> ", Inicio,"<br><b>Dívida Garantida:</b> ","R$",funcao\_formata\_quantidade\_2(total\_total)," Milhões","<br><b>Parcela Estados:</b> ",funcao\_percentual\_taxa(total\_total/sum(total\_total)) ,"<br><b>Parcela Total:</b> ",funcao\_percentual\_taxa(total\_total/sum(agrupador\_total\_formatado$total\_total)),"<br><b>Posição Estados:</b> ",order(total\_total, decreasing = TRUE),"° Lugar" ),

#"<br><b>Total Estados garantidos:</b> ",nrow(agrupador\_formatado\_estados)

hoverinfo = "text"

) %>%

#hide\_colorbar() %>%

layout(showlegend=TRUE, width = 880, height = 650 ,

legend = list(orientation = 'h', x = 0.9, y = 1,

xanchor = 'center', yref = 'paper',

font = list(size = 14),

bgcolor = 'transparent'),

title = "",

xaxis = list(title="", zeroline = FALSE,showline = FALSE , showgrid = FALSE), #categoryorder = "total ascending", categoryarray = "array"), showticklabels = FALSE

#rangeslider = list(type = "date")),

yaxis = list(side = 'left' ,title = '<b>Operações Garantidas - R$ Milhões</b> ', showline = FALSE ,showgrid = FALSE, zeroline = FALSE,tickformat = 'r',hoverformat = '.2f',

rangemode = "tozero"

)) %>%

config(displayModeBar = FALSE)

#layout(hovermode = 'compare')

)

```

> Fonte:

<br>Nota: Todos os 26 estados mais o Distrito Federal possuem operações garantidas pela União.

### Custo Médio

```{r custos estados, warning=FALSE, message=FALSE }

plotlyOutput("custos\_estados")

cor\_reativo\_custo\_estados <- reactive({

list(input$Mutuario)

})

cores\_custo\_estados<-reactive({ifelse(agrupador\_custo\_estados\_formatado[,1]==cor\_reativo\_custo\_estados(),"#cccc00","#b3b3b3")})

output$custos\_estados <-renderPlotly(

plot\_ly(agrupador\_custo\_estados\_formatado, x=~Inicio) %>%

add\_trace(y = ~Custo\_total\_formatado,

type = 'bar',

#size=~log10(ATM\_Total),

#color = ~ATM\_Total,

#colors = c("#ffcccc","#ff6666",'#ff0000'),

marker = list (color = cores\_custo\_estados()), #sizemode = 'diameter' #line = list(color = '#b3b3b3'opacity = 0.5

text = ~paste0("<b>Estado:</b> ", Inicio,"<br><b>Custo:</b> ",round(Custo\_total\_formatado,2),"% a.a","<br><b>Posição Estados:</b> ",rank(-Custo\_total\_formatado),"° Lugar","<br><b>Média Estados:</b> ",round(sum((agrupador\_custo\_estados\_formatado$Custo\_total\_formatado\*agrupador\_formatado\_estados$total\_total))/sum(agrupador\_formatado\_estados$total\_total),2),"% a.a" ),

hoverinfo = "text"

) %>%

hide\_colorbar() %>%

layout(autosize = F, width = 870, height = 650 ,legend = list(orientation = 'h', x = 0.8, y = 1.2,

xanchor = 'center', yref = 'paper',

font = list(size = 14),

bgcolor = 'transparent'),

title = "",

xaxis = list(title="", zeroline = FALSE,showline = FALSE ,showlegend = FALSE, showgrid = FALSE), #categoryorder = "total ascending", categoryarray = "array"), showticklabels = FALSE

#rangeslider = list(type = "date")),

yaxis = list(side = 'left',showlegend = FALSE ,title = '<b>Custo Médio - % a.a</b> ', showline = FALSE ,showgrid = FALSE, zeroline = FALSE,hoverformat = '.2f',

rangemode = "tozero"

)) %>%

config(displayModeBar = FALSE)

#layout(hovermode = 'compare')

)

```

> Fonte:

### Vida Média - ATM

```{r ATM, warning=FALSE, message=FALSE }

plotlyOutput("ATM")

cor\_reativo\_atm\_estados <- reactive({

list(input$Mutuario)

})

cores\_atm\_estados<-reactive({ifelse(agrupador\_ATM\_estados\_formatado[,1]==cor\_reativo\_atm\_estados(),"#cccc00","#b3b3b3")})

output$ATM <-renderPlotly(

plot\_ly(agrupador\_ATM\_estados\_formatado, x=~Inicio) %>%

add\_trace(y = ~ATM\_Total,

type = 'bar',

#size=~log10(ATM\_Total),

#color = ~ATM\_Total,

#colors = c("#ffcccc","#ff6666",'#ff0000'),

marker = list (color = cores\_atm\_estados()),#,line = list(color = '#0571b0', width =2 ) #sizemode = 'diameter'

text = ~paste0("<b>Estado:</b> ", Inicio,"<br><b>Vida Média:</b> ",round(ATM\_Total,2)," anos","<br><b>Posição Estados:</b> ",rank(-ATM\_Total),"° Lugar","<br><b>Média Estados:</b> ",round(sum((agrupador\_ATM\_estados\_formatado$ATM\_Total\*agrupador\_formatado\_estados$total\_total))/sum(agrupador\_formatado\_estados$total\_total),2)," anos"),

hoverinfo = "text"

) %>%

hide\_colorbar() %>%

layout(autosize = F, width = 870, height = 650 ,legend = list(orientation = 'h', x = 0.8, y = 1.2,

xanchor = 'center', yref = 'paper',

font = list(size = 14),

bgcolor = 'transparent'),

title = "",

xaxis = list(title="", zeroline = TRUE,showline = FALSE ,showlegend = FALSE, showgrid = FALSE), #categoryorder = "total ascending", categoryarray = "array"), showticklabels = FALSE

#rangeslider = list(type = "date")),

yaxis = list(side = 'left',showlegend = FALSE,showgrid = FALSE ,title = '<b>Vida Média - ATM (anos)</b> ', showline = FALSE , zeroline = FALSE,hoverformat = '.2f',

rangemode = "tozero"

)) %>%

config(displayModeBar = FALSE)

#layout(hovermode = 'compare')

)

```

> Fonte:

### Operações Garantidas Cambiais

```{r cambial estados, warning=FALSE, message=FALSE }

plotlyOutput("cambial\_estados")

cor\_reativo\_cambial\_estados <- reactive({

list(input$Mutuario)

})

cores\_cambial\_estados<-reactive({ifelse(agrupador\_custo\_estados\_formatado[,1]==cor\_reativo\_cambial\_estados(),"#cccc00","#b3b3b3")})

output$cambial\_estados <-renderPlotly(

plot\_ly(agrupador\_formatado\_estados, x=~Inicio) %>%

add\_trace(y = ~100\*((total\_cambial)/(total\_total)),

type = 'bar',

#size=~log10(ATM\_Total),

#color = ~ATM\_Total,

#colors = c("#ffcccc","#ff6666",'#ff0000'),

marker = list (color = cores\_cambial\_estados()), #sizemode = 'diameter' #line = list(color = '#b3b3b3'

text = ~paste0("<b>Estado:</b> ", Inicio,"<br><b>Peso Cambial:</b> ",round(100\*(total\_cambial/total\_total),2),"%","<br><b>Posição Estados:</b> ",rank(-total\_cambial/total\_total),"° Lugar","<br><b>Média Estados:</b> ",100\*round(sum(((total\_cambial/total\_total)\*agrupador\_formatado\_estados$total\_total))/sum(agrupador\_formatado\_estados$total\_total),2),"%" ),

hoverinfo = "text"

) %>%

hide\_colorbar() %>%

layout(autosize = F ,width = 870, height = 650 ,legend = list(orientation = 'h', x = 0.8, y = 1.2,

xanchor = 'center', yref = 'paper',

font = list(size = 14),

bgcolor = 'transparent'),

title = "",

xaxis = list(title="", zeroline = TRUE,showline = FALSE ,showlegend = FALSE, showgrid = FALSE), #categoryorder = "total ascending", categoryarray = "array"), showticklabels = FALSE

#rangeslider = list(type = "date")),

yaxis = list(side = 'left',showlegend = FALSE ,title = '<b>Operações Garantidas Cambiais - Peso (%)</b> ', showline = FALSE ,showgrid = FALSE, zeroline = FALSE,hoverformat = '.2f',

rangemode = "tozero"

)) %>%

config(displayModeBar = FALSE)

#layout(hovermode = 'compare')

)

```

> Fonte: .

<br>Dívida Cambial é a soma entre a dívida interna cambinal mais a dívida externa. O peso da dívida cambinal na carteira é calculado pela divisão entre a parcela cambial e a total.

Municípios

=======================================================================

Column

-----------------------------------------------------------------------

### {.card}

```{r Municipios, warning=FALSE, message=FALSE }

#Municipios - BASE DE DADOS: formatar e filtrar

agrupador\_formatado\_municipios <- agrupador\_total %>%

filter(!(Inicio %in% grupos)) %>%

filter((Classificador == "Municípios")) %>%

mutate(total\_total = numero(total\_total), interna\_total = numero(interna\_total), interna\_USD = numero(interna\_USD), externa\_total = numero(externa\_total), total\_cambial = numero(total\_cambial)) %>%

mutate(media\_total\_municipios = mean(total\_total))

agrupador\_formatado\_municipios$total\_total[agrupador\_formatado\_municipios$total\_total==0] <- NA

agrupador\_formatado\_municipios <- agrupador\_formatado\_municipios %>%

filter(!is.na(total\_total))

agrupador\_ATM\_municipios\_formatado <- agrupador\_atm\_completo %>%

filter(!(Inicio %in% grupos)) %>%

filter((Classificador == "Municípios")) %>%

mutate(ATM\_Total = numero(ATM\_Total))

agrupador\_custo\_municipios\_formatado <- agrupador\_custo\_completo %>%

filter(!(Inicio %in% grupos)) %>%

filter((Classificador == "Municípios")) %>%

mutate(Custo\_total\_formatado = numero(str\_replace\_all(Custo\_Total,"%","")))

```

```{r Seletor\_Municipio, warning=FALSE, message=FALSE }

fluidPage(

fluidRow(

column(5,div(selectInput("Municipios",

label="",

choices = agrupador\_formatado\_municipios$Inicio), id='seletor'))

))

botoes <- reactive({

list(input$Municipios)

})

observeEvent(botoes(),{

if (is.null(input$Municipios)){entrada <- agrupador\_formatado\_municipios$Inicio[1]}

else

{entrada <- input$Municipios}

if(entrada != "all"){

filtro\_municipios <- agrupador\_formatado\_municipios %>%

filter(Inicio == entrada) }

})

```

```{r card\_municipios, warning=FALSE, message=FALSE }

fluidPage(

fluidRow(

br(),

br(), br(),

column(width=6,align="center",div(style = "height:17px;font-size: 12pt;background-color: ;","")),

column(width=3,align="left",div(style = "height:17px;font-size: 12pt;background-color: ;","R$ Milhões")),

br(),

column(width=6,align="left",div(style = "height:17px;font-size: 13pt;font-weight: bold;background-color: ;","OPERAÇÕES GARANTIDAS")),

column(width=2,align="right",div(style = "height:17px;font-size: 13pt;font-weight: bold;background-color: ;",textOutput("Garantia\_Total\_municipios"))),

br(),

column(width=6,align="left",div(style = "height:17px;font-size: 13pt;background-color: ;","INTERNAS")),

column(width=2,align="right",div(style = "height:17px;font-size: 13pt;background-color: ;",textOutput("Garantia\_interna\_municipios"))),

br(),

column(width=1,align="left",div(style = "height:17px;font-size: 12pt;background-color: ;","")),

column(width=5,align="center",div(style = "height:17px;color: #737373;background-color: ;","Internas Cambiais:")),

column(width=2,align="right",div(style = "height:17px;color: #737373;font-size: 13pt;background-color: ;",textOutput("Garantia\_interna\_cambial\_municipios"))),

br(),

column(width=1,align="right",div(style = "height:17px;font-size: 12pt;background-color: ;","")),

column(width=4,align="right",div(style = "height:17px;color: #737373;background-color: ;","Internas Demais:")),

column(width=1,align="right",div(style = "height:17px;font-size: 12pt;background-color: ;","")),

column(width=2,align="right",div(style = "height:17px;color: #737373;font-size: 13pt;background-color: ;",textOutput("Garantia\_interna\_demais\_municipios"))),

br(),

column(width=6,align="left",div(style = "height:17px;font-size: 13pt;background-color: ;","EXTERNAS")),

column(width=2,align="right",div(style = "height:17px;font-size: 13pt;background-color: ;",textOutput("Garantia\_externa\_municipios"))),

br(), br(),

column(width=4,align="left",div(style = "height:17px;font-size: 13pt;font-weight: bold;background-color: ;","VIDA MÉDIA - ATM")),

column(width=4,align="right",div(style = "height:17px;font-size: 13pt;font-weight: bold;background-color: ;",textOutput("Garantia\_ATM\_municipios"))),

br(),br(),

column(width=4,align="left",div(style = "height:17px;font-size: 13pt;font-weight: bold;background-color: ;","CUSTO MÉDIO")),

column(width=4,align="right",div(style = "height:17px;font-size: 13pt;font-weight: bold;background-color: ;",textOutput("Garantia\_custo\_municipios"))),

br(),br(),

column(width=6,align="left",div(style = "height:17px;font-size: 13pt;font-weight: bold;background-color: ;","PERCENTUAL VINCENDO")),

column(width=2,align="center",div(style = "height:17px;font-size: 12pt;background-color: ;","")),

column(width=3,align="right",div(style = "height:17px;font-size: 12pt;background-color: ;","R$ Milhões")),

br(),

column(width=4,align="left",div(style = "height:17px;background-color: ;","Até 12 meses")),

column(width=4,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_12\_meses\_percentual\_municipios"))),

column(width=3,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_12\_meses\_municipios"))),

br(),

column(width=4,align="left",div(style = "height:17px;background-color: ;","De 1 a 2 anos")),

column(width=4,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_1\_ano\_percentual\_municipios"))),

column(width=3,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_1\_ano\_municipios"))),

br(),

column(width=4,align="left",div(style = "height:17px;background-color: ;","De 2 a 3 anos")),

column(width=4,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_2\_ano\_percentual\_municipios"))),

column(width=3,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_2\_ano\_municipios"))),

br(),

column(width=4,align="left",div(style = "height:17px;background-color: ;","De 3 a 4 anos")),

column(width=4,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_3\_ano\_percentual\_municipios"))),

column(width=3,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_3\_ano\_municipios"))),

br(),

column(width=4,align="left",div(style = "height:17px;background-color: ;","De 4 a 5 anos")),

column(width=4,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_4\_ano\_percentual\_municipios"))),

column(width=3,align="right",div(style = "height:17px;background-color: ;",textOutput("Garantia\_4\_ano\_municipios"))),

br(),

column(width=4,align="left",div(style = "height:17px;background-color: ;","Mais de 5 anos")),

column(width=4,align="right",div(style = "height:17px;;background-color: ;",textOutput("Garantia\_5\_ano\_percentual\_municipios"))),

column(width=3,align="right",div(style = "height:17px;;background-color: ;",textOutput("Garantia\_5\_ano\_municipios")))

))

#REATIVOS

Garantia\_Total\_reativo\_municipios <- reactive({

list(input$Municipios)

})

Garantia\_ATM\_reativo\_municipios <- reactive({

list(input$Municipios)

})

Garantia\_Custo\_reativo\_municipios <- reactive({

list(input$Municipios)

})

Garantia\_Vincendo\_reativo\_municipios <- reactive({

list(input$Municipios)

})

observeEvent(Garantia\_Total\_reativo\_municipios(),{

if (is.null(input$Municipios)){entrada <- "agrupador\_total$Classificador[1]"}

else

{entrada <- input$Municipios}

if(entrada != "all"){

filtro\_classificador\_municipios <- agrupador\_formatado\_municipios %>%

filter(Inicio == entrada) }

#Garantia Total, Interna Total, Interna Cambinal, Interna Demais, Externa

output$Garantia\_Total\_municipios <- renderText(paste0(funcao\_formata\_quantidade(filtro\_classificador\_municipios$total\_total)))

output$Garantia\_interna\_municipios <- renderText(paste0(funcao\_formata\_quantidade(filtro\_classificador\_municipios$interna\_total)))

output$Garantia\_interna\_cambial\_municipios <- renderText(paste0(funcao\_formata\_quantidade(filtro\_classificador\_municipios$interna\_USD)))

output$Garantia\_interna\_demais\_municipios <- renderText(paste0(funcao\_formata\_quantidade(filtro\_classificador\_municipios$interna\_total-filtro\_classificador\_municipios$interna\_USD)))

output$Garantia\_externa\_municipios <- renderText(paste0(funcao\_formata\_quantidade(filtro\_classificador\_municipios$externa\_total)))

})

#ATM - Municipios

observeEvent(Garantia\_ATM\_reativo\_municipios(),{

if (is.null(input$Municipios)){entrada <- "agrupador\_atm\_completo$Classificador[1]"}

else

{entrada <- input$Municipios}

if(entrada != "all"){

filtro\_classificador\_estado\_atm\_municipios <- agrupador\_ATM\_municipios\_formatado %>%

filter(Inicio == entrada) }

output$Garantia\_ATM\_municipios <- renderText(paste0(str\_replace\_all(round(filtro\_classificador\_estado\_atm\_municipios$ATM\_Total,2),"\\.",","), " anos"))

})

#CUSTO Médio

observeEvent(Garantia\_Custo\_reativo\_municipios(),{

if (is.null(input$Municipios)){entrada <- "agrupador\_custo\_completo$Classificador[1]"}

else

{entrada <- input$Municipios}

if(entrada != "all"){

filtro\_classificador\_custo\_municipios <- agrupador\_custo\_municipios\_formatado %>%

filter(Inicio == entrada) }

output$Garantia\_custo\_municipios <- renderText(paste0(filtro\_classificador\_custo\_municipios$Custo\_Total, " a.a"))

})

#PERCENTUAL VINCENDO

observeEvent(Garantia\_Vincendo\_reativo\_municipios(),{

if (is.null(input$Municipios)){entrada <- "agrupador\_custo\_completo$Classificador[1]"}

else

{entrada <- input$Municipios}

if(entrada != "all"){

filtro\_classificador\_vincendo\_municipios <- agrupador\_percentual\_vincendo %>%

filter(Classificador == "Municípios") %>%

filter(Inicio == entrada) }

output$Garantia\_12\_meses\_municipios <- renderText(paste0(funcao\_formata\_quantidade(numero(filtro\_classificador\_vincendo\_municipios$Ate\_12\_meses))))

output$Garantia\_12\_meses\_percentual\_municipios <- renderText(paste0(filtro\_classificador\_vincendo\_municipios$Ate\_12\_meses\_percentual))

output$Garantia\_1\_ano\_municipios <- renderText(paste0(funcao\_formata\_quantidade(numero(filtro\_classificador\_vincendo\_municipios$De\_1\_2\_anos))))

output$Garantia\_1\_ano\_percentual\_municipios <- renderText(paste0(filtro\_classificador\_vincendo\_municipios$De\_1\_2\_anos\_percentual))

output$Garantia\_2\_ano\_municipios <- renderText(paste0(funcao\_formata\_quantidade(numero(filtro\_classificador\_vincendo\_municipios$De\_2\_3\_anos))))

output$Garantia\_2\_ano\_percentual\_municipios <- renderText(paste0(filtro\_classificador\_vincendo\_municipios$De\_2\_3\_anos\_percentual))

output$Garantia\_3\_ano\_municipios <- renderText(paste0(funcao\_formata\_quantidade(numero(filtro\_classificador\_vincendo\_municipios$De\_3\_4\_anos))))

output$Garantia\_3\_ano\_percentual\_municipios <- renderText(paste0(filtro\_classificador\_vincendo\_municipios$De\_3\_4\_anos\_percentual))

output$Garantia\_4\_ano\_municipios <- renderText(paste0(funcao\_formata\_quantidade(numero(filtro\_classificador\_vincendo\_municipios$De\_4\_5\_anos))))

output$Garantia\_4\_ano\_percentual\_municipios <- renderText(paste0(filtro\_classificador\_vincendo\_municipios$De\_4\_5\_anos\_percentual))

output$Garantia\_5\_ano\_municipios <- renderText(paste0(funcao\_formata\_quantidade(numero(filtro\_classificador\_vincendo\_municipios$Acima\_5\_anos))))

output$Garantia\_5\_ano\_percentual\_municipios <- renderText(paste0(filtro\_classificador\_vincendo\_municipios$Acima\_5\_anos\_percentual))

})

```

### Contratos Assinados no \*\*ano\*\* {data-height=130}

```{r Tabela\_municipios, echo = FALSE, results = 'asis'}

htmlOutput("Tabela\_Municipio")

Tabela\_reativo\_municipios <- reactive({

list(input$Municipios)

})

observeEvent(Tabela\_reativo\_municipios(),{

if (is.null(input$Municipios)){entrada <- "agrupador\_custo\_completo$Classificador[1]"}

else

{entrada <- input$Municipios}

if(entrada != "all"){

filtro\_classificador\_municipios\_tabela <- novos\_contratos %>%

filter(`Tipo Mutuário` == "Municípios") %>%

mutate(`Valor` = paste0("R$ ",funcao\_formata\_quantidade(numero(`Valor Contratado Original`))," Milhões" )) %>%

filter(`Mutuário` == entrada) }

output$Tabela\_Municipio <- renderText({

kable(filtro\_classificador\_municipios\_tabela[c(17,3,9,10,26)], align = 'c') %>%

kable\_styling() %>%

scroll\_box(width = "100%", height = "135px")

})

})

```

Column {.tabset}

-----------------------------------------------------------------------

### Operações Garantidas

```{r divida\_total\_municipios, warning=FALSE, message=FALSE ,fig.width=9, fig.height=7 }

plotlyOutput("divida\_total\_municipios")

cor\_reativo\_divida\_total\_municipios <- reactive({

list(input$Municipios)

})

cores\_divida\_total\_municipios<-reactive({ifelse(agrupador\_formatado\_municipios[,1]==cor\_reativo\_divida\_total\_municipios(),"#cccc00","#b3b3b3")})

output$divida\_total\_municipios <-renderPlotly(

plot\_ly(agrupador\_formatado\_municipios, x=~Inicio) %>%

add\_trace(y = ~media\_total\_municipios/1000000 ,

type = 'scatter',

mode = 'lines',

line = list(color = '#8c8c8c', width = 1),

showlegend =TRUE,

visible = "legendonly",

name ="Média",

text = ~paste0("<b>Média:</b> ","R$ ",funcao\_formata\_quantidade(mean(total\_total))," Mihlões"),

hoverinfo = "text"

) %>%

add\_trace(y = ~(total\_total/1000000),

type = 'scatter',

size=~total\_total,

showlegend = FALSE,

marker = list (color = cores\_divida\_total\_municipios(), sizemode = 'diameter',

line = list(color = cores\_divida\_total\_municipios())),

text = ~paste0("<b>Município:</b> ", Inicio,"<br><b>Dívida Garantida:</b> ","R$",funcao\_formata\_quantidade\_2(total\_total)," Milhões","<br><b>Parcela Municípios:</b> ",funcao\_percentual\_taxa(total\_total/sum(total\_total)),"<br><b>Posição Municípios:</b> ",order(total\_total, decreasing = TRUE),"° Lugar" ),

#"<br><b>Total Municípios garantidos:</b> ",nrow(agrupador\_formatado\_municipios)

hoverinfo = "text"

) %>%

layout(showlegend=TRUE, width = 950, height = 650 ,

legend = list(orientation = 'h', x = 0.9, y = 1,

xanchor = 'center', yref = 'paper',

font = list(size = 14),

bgcolor = 'transparent'),

title = "",

xaxis = list(title="", zeroline = FALSE,showline = FALSE ,showlegend = FALSE, showgrid = FALSE, showticklabels = FALSE), #categoryorder = "total ascending", categoryarray = "array"), showticklabels = FALSE

#rangeslider = list(type = "date")),

yaxis = list(side = 'left',zeroline = FALSE,showlegend = FALSE ,title = '<b>Operações Garantidas - R$ Milhões</b> ', showline = FALSE ,showgrid = FALSE, zeroline = TRUE,tickformat = 'r',hoverformat = '.2f'

)) %>%

config(displayModeBar = FALSE)

#layout(hovermode = 'compare')

)

```

> \*\*Fonte:\*\*

<br>\*\*Nota:\*\* `r nrow(agrupador\_formatado\_municipios)` municípios possuem operações de crédito garantidas pela União.No entanto, `r nrow(agrupador\_formatado\_municipios)-max(rank(-agrupador\_ATM\_municipios\_formatado$ATM\_Total))` municípios ainda não relizaram nenhum desembolso e, portanto, seu saldo de operações garantidas é zero.

### Custo Médio

```{r custos\_municipios, warning=FALSE, message=FALSE }

plotlyOutput("custos\_municipios")

cor\_reativo\_custo\_municipios <- reactive({

list(input$Municipios)

})

cores\_custo\_municipios<-reactive({ifelse(agrupador\_custo\_municipios\_formatado[,1]==cor\_reativo\_custo\_municipios(),"#cccc00","#b3b3b3")})

output$custos\_municipios <-renderPlotly(

plot\_ly(agrupador\_custo\_municipios\_formatado, x=~Inicio) %>%

add\_trace(y = ~Custo\_total\_formatado,

type = 'scatter',

size=~Custo\_total\_formatado,

marker = list (color = cores\_custo\_municipios(),

line = list(color = cores\_custo\_municipios())),

text = ~paste0("<b>Município:</b> ", Inicio,"<br><b>Custo Médio:</b> ",round(Custo\_total\_formatado,2),"% a.a","<br><b>Posição Municípios:</b> ",rank(-Custo\_total\_formatado),"° Lugar","<br><b>Média Municípios:</b> ",round(sum((agrupador\_custo\_municipios\_formatado$Custo\_total\_formatado\*agrupador\_formatado\_municipios$total\_total))/sum(agrupador\_formatado\_municipios$total\_total),2),"% a.a" ),

hoverinfo = "text"

) %>%

layout(autosize = F, width = 900, height = 650 ,legend = list(orientation = 'h', x = 0.8, y = 1.2,

xanchor = 'center', yref = 'paper',

font = list(size = 14),

bgcolor = 'transparent'),

title = "",

xaxis = list(title="", zeroline = FALSE,showline = FALSE ,showlegend = FALSE, showgrid = FALSE, showticklabels = FALSE), #categoryorder = "total ascending", categoryarray = "array"), showticklabels = FALSE

#rangeslider = list(type = "date")),

yaxis = list(side = 'left',showlegend = FALSE ,title = '<b>Custo Médio - % a.a</b> ', showline = FALSE ,showgrid = FALSE, zeroline = FALSE,hoverformat = '.2f'

)) %>%

config(displayModeBar = FALSE)

#layout(hovermode = 'compare')

)

```

> Fonte:

<br>\*\*Nota:\*\* `r nrow(agrupador\_formatado\_municipios)-max(rank(-agrupador\_ATM\_municipios\_formatado$ATM\_Total))` municípios ainda não relizaram nenhum desembolso e, portanto, não há informações sobre seus respectivos custos.

### Vida Média - ATM

```{r ATM\_municipios, warning=FALSE, message=FALSE }

plotlyOutput("ATM\_municipios")

cor\_reativo\_atm\_municipios <- reactive({

list(input$Municipios)

})

cores\_atm\_municipios<-reactive({ifelse(agrupador\_ATM\_municipios\_formatado[,1]==cor\_reativo\_atm\_municipios(),"#cccc00","#b3b3b3")})

output$ATM\_municipios <-renderPlotly(

plot\_ly(agrupador\_ATM\_municipios\_formatado, x=~Inicio) %>%

add\_trace(y = ~ATM\_Total,

type = 'scatter',

size=~ATM\_Total,

marker = list (color = cores\_atm\_municipios(),

line = list(color = cores\_atm\_municipios())),

text = ~paste0("<b>Estado:</b> ", Inicio,"<br><b>Vida Média:</b> ",round(ATM\_Total,2)," anos","<br><b>Posição Estados:</b> ",rank(-ATM\_Total),"° Lugar","<br><b>Média Estados:</b> ",round(sum((agrupador\_ATM\_estados\_formatado$ATM\_Total\*agrupador\_formatado\_estados$total\_total))/sum(agrupador\_formatado\_estados$total\_total),2)," anos"),

hoverinfo = "text"

) %>%

layout(autosize = F, width = 900, height = 650 ,legend = list(orientation = 'h', x = 0.8, y = 1.2,

xanchor = 'center', yref = 'paper',

font = list(size = 14),

bgcolor = 'transparent'),

title = "",

xaxis = list(title="", zeroline = TRUE,showline = FALSE ,showlegend = FALSE, showgrid = FALSE, showticklabels = FALSE), #categoryorder = "total ascending", categoryarray = "array"), showticklabels = FALSE

#rangeslider = list(type = "date")),

yaxis = list(side = 'left',showlegend = FALSE,showgrid = FALSE ,title = '<b>Vida Média - ATM (anos)</b> ', showline = FALSE , zeroline = FALSE,hoverformat = '.2f'

)) %>%

config(displayModeBar = FALSE)

#layout(hovermode = 'compare')

)

```

> \*\*Fonte:\*\*

<br>\*\*Nota:\*\* `r nrow(agrupador\_formatado\_municipios)-max(rank(-agrupador\_ATM\_municipios\_formatado$ATM\_Total))` municípios ainda não relizaram nenhum desembolso e, portanto, não há informações sobre suas respectivos vidas médias (ATM).

### Operações Garantidas Cambiais

```{r cambial\_municipios, warning=FALSE, message=FALSE }

plotlyOutput("cambial\_municipios")

cor\_reativo\_cambial\_municipios <- reactive({

list(input$Municipios)

})

cores\_cambial\_municipios<-reactive({ifelse(agrupador\_formatado\_municipios[,1]==cor\_reativo\_cambial\_municipios(),"#cccc00","#b3b3b3")})

output$cambial\_municipios <-renderPlotly(

plot\_ly(agrupador\_formatado\_municipios, x=~Inicio) %>%

add\_trace(y = ~100\*((total\_cambial)/(total\_total)),

type = 'scatter',

size=~100\*((total\_cambial)/(total\_total)),

marker = list (color = cores\_cambial\_municipios(),

line = list(color = cores\_cambial\_municipios())),

text = ~paste0("<b>Municípios:</b> ", Inicio,"<br><b>Peso Cambial:</b> ",round(100\*(total\_cambial/total\_total),2),"%","<br><b>Posição Municípios:</b> ",rank(-total\_cambial/total\_total),"° Lugar","<br><b>Média Municípios:</b> ",100\*round(sum(((total\_cambial/total\_total)\*agrupador\_formatado\_municipios$total\_total))/sum(agrupador\_formatado\_municipios$total\_total),2),"%" ),

hoverinfo = "text"

) %>%

hide\_colorbar() %>%

layout(autosize = F ,width = 900, height = 650 ,legend = list(orientation = 'h', x = 0.8, y = 1.2,

xanchor = 'center', yref = 'paper',

font = list(size = 14),

bgcolor = 'transparent'),

title = "",

xaxis = list(title="", zeroline = TRUE,showline = FALSE ,showlegend = FALSE, showgrid = FALSE, showticklabels = FALSE), #categoryorder = "total ascending", categoryarray = "array"), showticklabels = FALSE

#rangeslider = list(type = "date")),

yaxis = list(side = 'left',showlegend = FALSE ,title = '<b>Operações Garantidas Cambiais - Peso (%)</b> ', showline = FALSE ,showgrid = FALSE, zeroline = FALSE,hoverformat = '.2f'

)) %>%

config(displayModeBar = FALSE)

#layout(hovermode = 'compare')

)

```

> Fonte: .

<br>Dívida Cambial é a soma entre a dívida interna cambinal mais a dívida externa. O peso da dívida cambinal na carteira é calculado pela divisão entre a parcela cambial e a total.

Estatais Federais

=======================================================================

Column

-----------------------------------------------------------------------

### SALDO GARANTIAS - Quadrmiestre xx

![](Imagens/estatais.png)

Column

-----------------------------------------------------------------------

### Detalhamento - Composição das Garantias

![](Imagens/opcao\_estatal.png)

![](Imagens/Card\_estatal.png)

Bancos Federais

=======================================================================

Column

-----------------------------------------------------------------------

### SALDO GARANTIAS - Quadrmiestre xx

![](Imagens/Bancos\_Federais.png)

Column

-----------------------------------------------------------------------

### Detalhamento - Composição das Garantias

![](Imagens/opcao\_municipio.png)

![](Imagens/Card\_RJ.png)

Entidades Controladas

=======================================================================

Column

-----------------------------------------------------------------------

### SALDO GARANTIAS - Quadrmiestre xx

![](Imagens/entidades.png)

Column

-----------------------------------------------------------------------

### Detalhamento - Composição das Garantias

![](Imagens/opcao\_municipio.png)

![](Imagens/Card\_RJ.png)

Quadro Geral

=======================================================================

Column

-------------------------------------------------------------------------

### SALDO GARANTIAS A OPERAÇÕES DE CRÉDITO (quadrimestre xx)

![](Imagens/correto.png)

Column

-------------------------------------------------------------------------

### MUTUÁRIOS - COMPOSIÇÃO DAS GARANTIAS (quadrimestre xx)

![](Imagens/mutuario\_geral.png)

Column

-----------------------------------------------------------------------

### Credores - COMPOSIÇÃO DAS GARANTIAS (quadrimestre xx)

![](Imagens/credor\_total.png)

Contratos Assinados

=======================================================================

Conceitos

=======================================================================