library(shiny)

library(shinydashboard)

ui<-dashboardPage(

dashboardHeader(),

dashboardSidebar(),

dashboardBody()

)

server<-function(input,output){}

shinyApp(ui=ui,server=server)

###################################################################################################

ui<-dashboardPage(

dashboardHeader(title="Orçamento"),

dashboardSidebar(),

dashboardBody()

)

server<-function(input,output){}

shinyApp(ui=ui,server=server)

###################################################################################################

ui<-dashboardPage(skin = "green",

dashboardHeader(title="Orçamento"),

dashboardSidebar(),

dashboardBody()

)

server<-function(input,output){}

shinyApp(ui=ui,server=server)

###################################################################################################

ui<-dashboardPage(skin = "green",

dashboardHeader(title="Bem vindo ao portal de transparência Orçamentária"),

# dashboardHeader(title="Bem vindo ao portal de transparência Orçamentária", titleWidth = 700),

dashboardSidebar(),

dashboardBody()

)

server<-function(input,output){}

shinyApp(ui=ui,server=server)

###################################################################################################

ui<-dashboardPage(skin = "green",

dashboardHeader(title="Orçamento", titleWidth = 700),

dashboardSidebar(),

dashboardBody()

)

server<-function(input,output){}

shinyApp(ui=ui,server=server)

###################################################################################################

ui<-dashboardPage(

dashboardHeader(title="Orçamento",

# Dropdown menu for messages

dropdownMenu(type = "messages", badgeStatus = "success",

messageItem("Support Team","This is the content of a message.",time = "5 mins"),

messageItem("Support Team","This is the content of another message.",time = "2 hours"),

messageItem("New User","Can I get some help?", time = "Today")

),

# Dropdown menu for notifications

dropdownMenu(type = "notifications", badgeStatus = "warning",

notificationItem(icon = icon("users"), status = "info","5 new members joined today" ),

notificationItem(icon = icon("warning"), status = "danger","Resource usage near limit."),

notificationItem(icon = icon("shopping-cart", lib = "glyphicon"),status = "success", "25 sales made"),

notificationItem(icon = icon("user", lib = "glyphicon"),status = "danger", "You changed your username")

),

# Dropdown menu for tasks, with progress bar

dropdownMenu(type = "tasks", badgeStatus = "danger",

taskItem(value = 20, color = "aqua","Refactor code"),

taskItem(value = 40, color = "green","Design new layout"),

taskItem(value = 60, color = "yellow","Another task"),

taskItem(value = 80, color = "red","Write documentation")

)

),

dashboardSidebar(),

dashboardBody()

)

server<-function(input,output){}

shinyApp(ui=ui,server=server)

###################################################################################################

ui<-dashboardPage(

dashboardHeader(title="Orçamento"),

dashboardSidebar(

h4("Gráficos"),

selectInput("size", "Size:",

list("Empenhado" = "Empenhado",

"Liquidado" = "Liquidado",

"Pago\_e\_RP\_Pago" = "Pago\_e\_RP\_Pago")),

selectInput("classe", "Classe",

list("Acao\_Subtitulo" = "Acao\_Subtitulo",

"Programa" = "Programa",

"Categoria\_Economica" = "Categoria\_Economica",

"Elemento\_Despesa" = "Elemento\_Despesa"))

),

dashboardBody()

)

server<-function(input,output){}

shinyApp(ui=ui,server=server)

###################################################################################################

ui<-dashboardPage(

dashboardHeader(title="Orçamento"),

dashboardSidebar(

h4("Gráficos"),

selectInput("size", "Size:",

list("Empenhado" = "Empenhado",

"Liquidado" = "Liquidado",

"Pago\_e\_RP\_Pago" = "Pago\_e\_RP\_Pago")),

selectInput("classe", "Classe",

list("Acao\_Subtitulo" = "Acao\_Subtitulo",

"Programa" = "Programa",

"Categoria\_Economica" = "Categoria\_Economica",

"Elemento\_Despesa" = "Elemento\_Despesa"))

),

dashboardBody(

tabsetPanel(

tabPanel("Gráfico",value=1,

fluidRow(),

h5("Fonte dos dados: SIAFI."))

)

)

)

server<-function(input,output){}

shinyApp(ui=ui,server=server)

###################################################################################################

ui<-dashboardPage(

dashboardHeader(title="Orçamento"),

dashboardSidebar(

h4("Gráficos"),

selectInput("size", "Size:",

list("Empenhado" = "Empenhado",

"Liquidado" = "Liquidado",

"Pago\_e\_RP\_Pago" = "Pago\_e\_RP\_Pago")),

selectInput("classe", "Classe",

list("Acao\_Subtitulo" = "Acao\_Subtitulo",

"Programa" = "Programa",

"Categoria\_Economica" = "Categoria\_Economica",

"Elemento\_Despesa" = "Elemento\_Despesa"))

),

dashboardBody(

tabsetPanel(

tabPanel("Gráfico",value=1,

fluidRow(

valueBoxOutput("Empenhado1"),

valueBoxOutput("Liquidado1"),

valueBoxOutput("Pago1")),

h5("Fonte dos dados: SIAFI.")),

tabPanel("Análise do Orçamento",value=2,

h2("Texto longo, porém interessante"),

p(strong("Empenho"),"é o ato emanado de autoridade competente que cria para o Estado obrigação de pagamento. Consiste na reserva de dotação orçamentária para um fim específico. Ressalta-se que não é possível a realização da despesa sem prévio empenho."),

h2(strong("Visite o nosso site: "))

)

)

)

)

server<-function(input,output){

setwd("C:/Rmarkdowneshiny-master/shinydashboard")

load(file="Transparencia2016.RData")

Empenhado <- sum(Transparencia2016$Empenhado)

Liquidado <- sum(Transparencia2016$Liquidado)

Pago\_e\_RP\_Pago <- sum(Transparencia2016$Pago\_e\_RP\_Pago)

output$Empenhado1 <- renderValueBox({valueBox(

value = Empenhado,

subtitle = "Empenhado em 2016", icon = icon("area-chart"),

color = if (Liquidado >= Empenhado) "yellow" else "aqua")

})

output$Liquidado1 <- renderValueBox({ valueBox(

value = Liquidado,

subtitle = "Liquidado em 2016",

icon = icon("line-chart"),

color = "yellow")

})

output$Pago1 <- renderValueBox({ valueBox(

value = Pago\_e\_RP\_Pago,

subtitle = "Pago e RP Pago em 2016",

icon = icon("university"),

color = "green")

})

}

shinyApp(ui=ui,server=server)

###################################################################################################

ui<-dashboardPage(

dashboardHeader(title="Orçamento"),

dashboardSidebar(

h4("Gráficos"),

selectInput("size", "Size:",

list("Empenhado" = "Empenhado",

"Liquidado" = "Liquidado",

"Pago\_e\_RP\_Pago" = "Pago\_e\_RP\_Pago")),

selectInput("classe", "Classe",

list("Acao\_Subtitulo" = "Acao\_Subtitulo",

"Programa" = "Programa",

"Categoria\_Economica" = "Categoria\_Economica",

"Elemento\_Despesa" = "Elemento\_Despesa"))

),

dashboardBody(

tabsetPanel(

tabPanel("Gráfico",value=1,

fluidRow(

valueBoxOutput("Empenhado1"),

valueBoxOutput("Liquidado1"),

valueBoxOutput("Pago1"),

plotOutput("plot")),

h5("Fonte dos dados: SIAFI.")),

tabPanel("Análise do Orçamento",value=2,

h2("Você sabe o que é um empenho?"),

p(strong("Empenho"),"é o ato emanado de autoridade competente que cria para o Estado obrigação de pagamento. Consiste na reserva de dotação orçamentária para um fim específico. Ressalta-se que não é possível a realização da despesa sem prévio empenho."),

h2(strong("Visite o nosso site: "))

)

)

)

)

server<-function(input,output){

setwd("C:/Rmarkdowneshiny-master/shinydashboard")

load(file="Transparencia2016.RData")

Empenhado <- sum(Transparencia2016$Empenhado)

Liquidado <- sum(Transparencia2016$Liquidado)

Pago\_e\_RP\_Pago <- sum(Transparencia2016$Pago\_e\_RP\_Pago)

output$Empenhado1 <- renderValueBox({valueBox(

value = paste0("R$ ",formatC(Empenhado, digits = 2, big.mark = ".", big.interval = 3L, small.mark = " .", decimal.mark = ",",format = "f")),

subtitle = "Empenhado em 2016", icon = icon("area-chart"),

color = if (Liquidado >= Empenhado) "yellow" else "aqua")

})

output$Liquidado1 <- renderValueBox({ valueBox(

value = paste0("R$ ",formatC(Empenhado, digits = 2, big.mark = ".", big.interval = 3L, small.mark = " .", decimal.mark = ",",format = "f")),

subtitle = "Liquidado em 2016",

icon = icon("line-chart"),

color = "yellow")

})

output$Pago1 <- renderValueBox({ valueBox(

value = paste0("R$ ",formatC(Empenhado, digits = 2, big.mark = ".", big.interval = 3L, small.mark = " .", decimal.mark = ",",format = "f")),

subtitle = "Pago e RP Pago em 2016",

icon = icon("university"),

color = "green")

})

output$plot <- renderPlot({

par(mar=c(0,0,0,0))

treemap(Transparencia2016,

index=input$classe,

vSize=input$size,

vColor = input$size)

})

}

shinyApp(ui=ui,server=server)

###################################################################################################

# Desafio: Onde está o erro na programação?

ui<-dashboardPage(

dashboardHeader(title="Orçamento"),

dashboardSidebar(

h4("Gráficos"),

selectInput("size", "Size:",

list("Empenhado" = "Empenhado",

"Liquidado" = "Liquidado",

"Pago\_e\_RP\_Pago" = "Pago\_e\_RP\_Pago")),

selectInput("classe", "Classe",

list("Acao\_Subtitulo" = "Acao\_Subtitulo",

"Programa" = "Programa",

"Categoria\_Economica" = "Categoria\_Economica",

"Elemento\_Despesa" = "Elemento\_Despesa"))

),

dashboardBody(

tabsetPanel(

tabPanel("Gráfico",value=1,

fluidRow(

valueBoxOutput("Empenhado1"),

valueBoxOutput("Liquidado1"),

valueBoxOutput("Pago1"),

valueBoxOutput("Inicial1"),

plotOutput("plot")),

br(),

br(),

br(),

br(),

br(),

h5("Fonte dos dados: SIAFI.")),

tabPanel("Análise do Orçamento",value=2,

h2("Você sabe o que é um empenho?"),

p(strong("Empenho"),"é o ato emanado de autoridade competente que cria para o Estado obrigação de pagamento. Consiste na reserva de dotação orçamentária para um fim específico. Ressalta-se que não é possível a realização da despesa sem prévio empenho."),

h2(strong("Visite o nosso site: "))),

tabPanel("Banco de Dados",value=3,

h2("Banco de dados"),

DT::dataTableOutput("data")),

tabPanel("Gráfico 2",value=4,

h2("Gráfico 2"),

plotOutput(outputId="Box"))

)

)

)

server<-function(input,output){

setwd("C:/Rmarkdowneshiny-master/shinydashboard")

load(file="Transparencia2016.RData")

Empenhado <- sum(Transparencia2016$Empenhado)

Liquidado <- sum(Transparencia2016$Liquidado)

Pago\_e\_RP\_Pago <- sum(Transparencia2016$Pago\_e\_RP\_Pago)

Dotacao\_Inicial <- sum(Transparencia2016$Dotacao\_Inicial)

output$Empenhado1 <- renderValueBox({valueBox(

value = formatC(Empenhado, digits = 2, big.mark = ".", big.interval = 3L, small.mark = " .", decimal.mark = ",",format = "f"),

subtitle = "Empenhado em 2016", icon = icon("area-chart"),

color = if (Liquidado >= Empenhado) "yellow" else "aqua")

})

output$Liquidado1 <- renderValueBox({ valueBox(

value = formatC(Liquidado , digits = 2, big.mark = ".", big.interval = 3L, small.mark = " .", decimal.mark = ",",format = "f"),

subtitle = "Liquidado em 2016",

icon = icon("line-chart"),

color = "yellow")

})

output$Pago1 <- renderValueBox({ valueBox(

value = formatC(Pago\_e\_RP\_Pago , digits = 2, big.mark = ".", big.interval = 3L, small.mark = " .", decimal.mark = ",",format = "f"),

subtitle = "Pago e RP Pago em 2016",

icon = icon("university"),

color = "green")

})

output$Inicial1 <- renderValueBox({ valueBox(

value = formatC(Dotacao\_Inicial , digits = 2, big.mark = ".", big.interval = 3L, small.mark = " .", decimal.mark = ",",format = "f"),

subtitle = "Dotação Inicial",

icon = icon("university"),

color = "yellow")

})

output$plot <- renderPlot({

par(mar=c(0,0,0,0))

treemap(Transparencia2016,

index=input$classe,

vSize=input$size,

vColor = input$size)

})

output$data = DT::renderDataTable({

Transparencia2016[]

})

output$Box<-renderPlot({

colm = as.numeric(Transparencia2016$Dotacao\_Inicial)

boxplot(Transparencia2016[,colm],col="blue",

xlab = names(Transparencia2016[colm]))})

}

shinyApp(ui=ui,server=server)

###################################################################################################

# Resposta

ui<-dashboardPage(

dashboardHeader(title="Orçamento"),

dashboardSidebar(

h4("Gráficos"),

selectInput("size", "Size:",

list("Empenhado" = "Empenhado",

"Liquidado" = "Liquidado",

"Pago\_e\_RP\_Pago" = "Pago\_e\_RP\_Pago")),

selectInput("classe", "Classe",

list("Acao\_Subtitulo" = "Acao\_Subtitulo",

"Programa" = "Programa",

"Categoria\_Economica" = "Categoria\_Economica",

"Elemento\_Despesa" = "Elemento\_Despesa"))

),

dashboardBody(

tabsetPanel(

tabPanel("Gráfico",value=1,

fluidRow(

valueBoxOutput("Empenhado1"),

valueBoxOutput("Liquidado1"),

valueBoxOutput("Pago1"),

plotOutput("plot")),

h5("Fonte dos dados: SIAFI.")),

tabPanel("Análise do Orçamento",value=2,

h2("Você sabe o que é um empenho?"),

p(strong("Empenho"),"é o ato emanado de autoridade competente que cria para o Estado obrigação de pagamento. Consiste na reserva de dotação orçamentária para um fim específico. Ressalta-se que não é possível a realização da despesa sem prévio empenho."),

h2(strong("Visite o nosso site: "))),

tabPanel("Banco de Dados",value=3,

h2("Banco de dados"),

tableOutput("data"))

)

)

)

server<-function(input,output){

setwd("C:/Users/Steven/Desktop/Rmarkdowneshiny/shinydashboard/")

load(file="Transparencia2016.RData")

Empenhado <- sum(Transparencia2016$Empenhado)

Liquidado <- sum(Transparencia2016$Liquidado)

Pago\_e\_RP\_Pago <- sum(Transparencia2016$Pago\_e\_RP\_Pago)

output$Empenhado1 <- renderValueBox({valueBox(

value = formatC(Empenhado, digits = 2, big.mark = ".", big.interval = 3L, small.mark = " .", decimal.mark = ",",format = "f"),

subtitle = "Empenhado em 2016", icon = icon("area-chart"),

color = if (Liquidado >= Empenhado) "yellow" else "aqua")

})

output$Liquidado1 <- renderValueBox({ valueBox(

value = formatC(Liquidado , digits = 2, big.mark = ".", big.interval = 3L, small.mark = " .", decimal.mark = ",",format = "f"),

subtitle = "Liquidado em 2016",

icon = icon("line-chart"),

color = "yellow")

})

output$Pago1 <- renderValueBox({ valueBox(

value = formatC(Pago\_e\_RP\_Pago , digits = 2, big.mark = ".", big.interval = 3L, small.mark = " .", decimal.mark = ",",format = "f"),

subtitle = "Pago e RP Pago em 2016",

icon = icon("university"),

color = "green")

})

output$plot <- renderPlot({

par(mar=c(0,0,0,0))

treemap(Transparencia2016,

index=input$classe,

vSize=input$size,

vColor = input$size)

})

output$data <- renderTable({

Transparencia2016[]

})

}

shinyApp(ui=ui,server=server)