

Apêndice E:**Código do aplicativo Simplex Journey Termos v.1.0.0**

Data de Publicação: 20 de março de 2025

```

library(shiny)
ui <-fluidPage(
  tags$head(
    tags$style(HTML("
      @import url('https://fonts.googleapis.com/css2?family=Yusei+Magic&display=swap');
      body {
        background-color: #b3b3ff;
        color: black;
      }
      h2 {
        font-family: 'Yusei Magic', sans-serif;
      }
      .shiny-input-container {
        color: #474747;
      }"))
  ),

  titlePanel("Simplex"),
  fluidRow(column(3,

    textInput("nome",
      "Como devo lhe chamar? Digite abaixo.",
      "Cadete"),
    sliderInput("n_termos", "Número de termos:",
      min = 3, max = 10, value = 5)

  ),
  column(9,
    h2(verbatimTextOutput("nome2"))
  )
),

navlistPanel(
  widths = c(2, 10),
  "Menu",
  tabPanel("Teste Conhecimento",
    icon = icon("pencil"),
    HTML("Teste seu conhecimento, uma frase aparecerá e você deve identificar o item correspondente"),
    tags$br(),
    HTML("O número de termos pode ser definido por você variando de 3 a 10, quanto mais termos melhor"),
    tags$br(),
    HTML("Depois verifique a resposta correta e veja se acertou!"),
    h3(verbatimTextOutput("instrucao1")),

    fluidPage(

      sidebarPanel(
        radioButtons("respostamarcada",
          "Radio Buttons input:",
          c("label 1" = "option1",
            "label 2" = "option2"),
          selected = NA),

```

```

    hr(),
    fluidRow(column(3, verbatimTextOutput("value")))
  ),

  mainPanel(
    h4("Frase"),
    uiOutput("explicacao"),
    h4(verbatimTextOutput("resposta")),
    h4(verbatimTextOutput("resultado")),
    selectInput("mostra_resposta", "Mostrar resposta?",
      choices = c("Não", "Sim")),

    conditionalPanel(
      condition = "input.mostra_resposta == 'Sim'",
      div(h4(verbatimTextOutput("respostacerta")))
    ),

    actionButton("nova_pergunta", "Gerar nova pergunta")
  )
)
)
)
)

library(shiny)
server = function(input, output, session) {
  termosglo=read.csv("termos.csv")
  termosexp=read.csv("termos_exp.csv")

  gerar_pergunta <- function(n_termos) {
    c_num = n_termos
    c_label = input$nome
    linhasorteada = sample(1:40,c_num) #numero das linhas sorteadas do glossario
    linhaescolhida = sample(linhasorteada,1) #numero da linha que corresponde a resposta certa
    options = termosglo$x[linhasorteada] #termos correspondentes às linhas sorteadas
    exp_num = termosexp$x[linhaescolhida]
    respostacerta = termosglo$x[linhaescolhida]

    # Set the label, choices, and selected item
    updateRadioButtons(session, "respostamarcada",
      label = paste0(c_label, ", leia a descrição e selecione"),
      choices = options,
      selected = character(0)
    )

    output$nome2 <- renderText({ paste("Bons estudos", input$nome, "!") })

    output$explicacao <- renderUI({
      HTML(exp_num)
    })

    output$instrucao1 <- renderText({ paste0(input$nome," , leia a frase e escolha o termo correto.") })
    output$resposta <- renderText({ paste0(input$nome," , sua resposta é: ", input$respostamarcada) })
    output$respostacerta <- renderText({ paste0(input$nome," , a resposta certa é: ", respostacerta) })

    output$resultado <- renderText({

```

```

    if (input$respostamarcada == respostacerta) {
      paste0(input$nome, ", sua resposta está correta!")
    } else {
      paste0(input$nome, ", infelizmente sua resposta está incorreta.")
    }
  })
}

observe({
  gerar_pergunta(input$n_termos)
})

observeEvent(input$nova_pergunta, {
  gerar_pergunta(input$n_termos)
})
}

shinyApp(ui, server)

```

Comentários Técnicos

1. Interface do Usuário (UI)

- **fluidPage:** Cria uma página fluida e responsiva.
- **tagsheadetagsstyle:** Adicionam estilos personalizados, incluindo a importação de uma fonte externa e a definição de cores.
- **titlePanel:** Define o título do aplicativo.
- **fluidRow e column:** Organizam os elementos da interface em uma grade. **textInput:** Campo de entrada de texto para o nome do usuário.
- **sliderInput:** Controle deslizante para selecionar o número de termos. **navlistPanel:** Cria um painel de navegação com abas.
- **tabPanel:** Define uma aba com conteúdo específico, incluindo instruções e controles de entrada.

2. Lógica do Servidor

- **server:** Define a função do servidor que contém a lógica do aplicativo.
- **read.csv:** Lê os arquivos CSV contendo os termos e suas explicações.
- **gerar__pergunta:** Função que gera uma nova pergunta com base no número de termos selecionados.
- **sample:** Seleciona aleatoriamente linhas do glossário.
- **updateRadioButtons:** Atualiza os botões de rádio com as opções de resposta.
- **renderText e renderUI:** Renderizam texto e HTML na interface do usuário.
- **observe:** Observa mudanças no número de termos e gera uma nova pergunta.
- **observeEvent:** Observa o evento de clique no botão “Gerar nova pergunta” e gera uma nova pergunta.
- **flexibilidade:** O uso de **sliderInput** para permitir que o usuário selecione o número de termos adiciona flexibilidade ao aplicativo, permitindo que ele se adapte a diferentes níveis de dificuldade.
- **Feedback Imediato:** O aplicativo fornece feedback imediato ao usuário sobre suas respostas, o que é crucial para o aprendizado. Mensagens como “sua resposta está correta” ou “infelizmente sua resposta está incorreta” ajudam o usuário a entender seu desempenho.