Opción B: Cargar proyecto existente (usando reactiveValues para guardar estado)

| **Paquete** | **Función principal** |
| --- | --- |
| shiny | Construcción de la app |
| shinydashboard o bslib | Para diseño/estructura moderna |
| reactable, DT, rhandsontable | Mostrar y editar tablas |
| readr, readxl, haven | Importar diferentes tipos de bases de datos |
| officer | Generar documentos Word y PowerPoint |
| flextable | Crear tablas elegantes dentro de los informes |
| rmarkdown | Alternativa para informes más flexibles |
| stringr, glue | Para manejar texto dinámico en los reportes |
| shinyFiles (opcional) | Si quieres permitir navegación por archivos del sistema |
| shinyWidgets, shinyjs | Para mejorar interactividad y diseño |

 Usar officer y flextable para generar el documento

 Botón downloadHandler() para permitir la descarga

**🧱 Estructura general & UI (interfaz de usuario)**

| **Paquete** | **Uso principal** |
| --- | --- |
| shiny | Estructura de la app web |
| shinydashboard o shiny.fluent | Crear tabs, menús y navegación más pro |
| shinyjs | Controlar interactividad y lógica condicional |
| shinyWidgets | Inputs más estéticos (sliders, checkboxes bonitos, modales) |
| shinyFiles | Subir, guardar y explorar archivos |
| shinyjqui / sortable | Para hacer **drag & drop** de bloques (como reordenar slides o secciones) |

📂 **Gestión de proyectos y datos**

| **Paquete** | **Uso** |
| --- | --- |
| readr, readxl, haven | Importación de bases de datos (CSV, Excel, SPSS, etc.) |
| janitor | Limpieza de nombres de variables y verificación rápida |
| dplyr, tidyr | Manipulación y transformación de datos |
| validate, pointblank | Validación de estructura, tipos, reglas sobre variables cargadas |

🔍 **Selección y análisis de datos**

| **Paquete** | **Uso** |
| --- | --- |
| reactable / DT | Mostrar tablas interactivas con filtros |
| esquisse | Selección y creación **guiada** de gráficos, sin necesidad de programar |
| ggplot2 | Creación de gráficos personalizados |
| ggiraph, plotly | Hacer los gráficos interactivos si lo deseas |
| gtsummary, summarytools | Para análisis estadísticos pre-armados (resúmenes, test, etc.) |
| tidymodels | Si quieres eventualmente incluir modelos predictivos |

🧾 **Generación de reportes (Word, PPT, PDF)**

| **Paquete** | **Uso** |
| --- | --- |
| officer | Creación de documentos Word (.docx) y PowerPoint (.pptx) desde R |
| flextable | Tablas bonitas y exportables con estilos (muy útil para Word) |
| rmarkdown | Alternativa si quieres usar plantillas de Word o PDF desde Rmd |
| rpptxbuilder (no oficial) | Explorar si quieres más control sobre diseño de PowerPoint |
| pagedown | Para PDF con diseño tipo presentación desde RMarkdown |
| webshot2 | Para capturas en PDF o imágenes (útil si incrustas visualizaciones) |

**💾 Guardar progreso / proyectos / sesiones**

| **Paquete** | **Uso** |
| --- | --- |
| pins | Guardar objetos (gráficos, tablas, datos) en caché local o en la nube |
| jsonlite, yaml | Serializar configuración de proyectos para cargarlos luego |
| RSQLite | Si quisieras llevar registro más estructurado de los proyectos en una base de datos interna |

🎨 **Identidad visual y branding**

| **Paquete** | **Uso** |
| --- | --- |
| bslib | Personalizar el tema visual de la app Shiny |
| sass, thematic | Integrar paletas de colores corporativos |
| magick | Manipular imágenes, como el logo de la empresa del cliente |
| showtext | Usar fuentes personalizadas (para documentos o gráficos) |

| **🧩 Etapa del flujo** | **🎯 Objetivo principal** | **📦 Paquetes recomendados** | **💡 Notas** |
| --- | --- | --- | --- |
| **Inicio del proyecto** | Crear nuevo proyecto / seleccionar uno ya iniciado | shiny, shinyFiles, jsonlite, RSQLite, pins | Guardar/cargar configuración, base de datos local |
| **Subida y validación de datos** | Importar y verificar estructura de datos | readr, readxl, haven, janitor, validate, pointblank | Limpieza automática, validación de tipos |
| **Selección de variables** | Elegir qué variables usar en análisis | shinyWidgets, reactable, DT, dplyr, tidyselect | Inputs dinámicos e interfaces amigables |
| **Análisis de datos** | Generar estadísticas, agrupaciones, resúmenes | dplyr, gtsummary, summarytools, tidymodels | Posibilidad de guardar análisis |
| **Visualización de datos** | Crear gráficos simples e interactivos | esquisse, ggplot2, plotly, ggiraph | esquisse permite gráficos sin código |
| **Guardar análisis** | Guardar gráficos/tablas seleccionados | pins, jsonlite, rds, magrittr | Guardar como objeto R para usar luego |
| **Diseño del documento** | Elegir formato, colores, logos, plantillas | officer, flextable, magick, showtext, bslib | Integrar identidad visual del cliente |
| **Construcción del informe** | Crear slides o secciones Word con análisis previos | officer, flextable, ggplot2, webshot2, shinyjqui, sortable | Arrastrar elementos guardados, generar documento |
| **Previsualización** | Ver estructura del informe antes de exportar | rmarkdown::render, officer::read\_docx, reactable | Opciones de preview y edición ligera |
| **Exportación final** | Descargar Word/PPT o PDF | officer, rmarkdown, webshot2, pagedown | Dependiendo del formato elegido |

# 📦 Estructura base de una Shiny App para reportes automatizados

library(shiny)

library(shinyWidgets)

ui <- fluidPage(

titlePanel("Generador de Reportes Automatizados"),

sidebarLayout(

sidebarPanel(

# Selector de proyecto

selectInput("project\_mode", "Seleccionar modo:",

choices = c("Nuevo Proyecto", "Cargar Proyecto"))

),

mainPanel(

tabsetPanel(

tabPanel("Inicio", value = "tab\_inicio"),

tabPanel("Subida y validación", value = "tab\_datos"),

tabPanel("Selección de variables", value = "tab\_variables"),

tabPanel("Análisis de datos", value = "tab\_analisis"),

tabPanel("Visualización de gráficos", value = "tab\_graficos"),

tabPanel("Diseño de reporte", value = "tab\_diseño"),

tabPanel("Previsualización", value = "tab\_preview"),

tabPanel("Descargar", value = "tab\_descarga")

)

)

)

)

server <- function(input, output, session) {

# Aquí irán los observers y lógica para cambiar entre tabs

# y renderizar cada parte del flujo según sea necesario

}

shinyApp(ui, server)