

# Informe Técnico: Funcionamiento de la Aplicación Shiny para Análisis Estadístico Interactivo

Autor: Quispe Cruz Joel Antoni

**Universidad Nacional del Altiplano**

Facultad de Ingeniería Estadística e Informática

Docente: Fred Torres Cruz

## Repositorio del código

<https://github.com/Joel-11-30/anova.git>

## 1. Objetivo General

La aplicación desarrollada en Shiny tiene como propósito permitir a los usuarios:

- Subir archivos de datos (.csv o .xlsx).
- Visualizar los datos cargados.
- Realizar análisis estadísticos descriptivos.
- Generar gráficos representativos.
- Aplicar pruebas estadísticas comunes.
- Explorar el Teorema del Límite Central mediante simulación.

## 2. Componentes Clave del Código

### 2.1. Carga de Paquetes y Estilo Visual

```
paquetes <- c("shiny", "readxl", "dplyr", "ggplot2",  
             "DescTools", "shinythemes", "report")
```

Se verifican e instalan los paquetes necesarios. Se usa el tema `slate` para una apariencia oscura profesional.

### 2.2. Interfaz de Usuario (UI)

#### 2.2.1. Subida de Datos

```
fileInput("archivo", "Carga tu archivo (.csv o .xlsx):",  
          accept = c(".csv", ".xlsx"))
```

#### 2.2.2. Selección Dinámica de Variables

```
uiOutput("seleccion_vars")
```

#### 2.2.3. Paneles de Navegación

```
navbarPage(...)
```

Estructura de pestañas para organizar las funcionalidades.

### 2.3. Lógica del Servidor (Server)

#### 2.3.1. Lectura Reactiva del Archivo

```
datos <- reactive({  
  req(input$archivo)  
  ...  
})
```

Lee los datos según el formato del archivo cargado.

#### 2.3.2. Vista Previa de Datos

```
output$vista_datos <- renderTable({ head(datos(), 10) })
```

#### 2.3.3. Estadísticas Descriptivas

```
output$estadisticas_ui <- renderUI({...})
```

Genera tablas según el tipo de variable: numérica o categórica.

### 2.3.4. Visualización Gráfica

```
output$graficos_ui <- renderUI({...})
```

Utiliza `ggplot2` para gráficos de barras e histogramas.

## 2.4. Pruebas Estadísticas

```
prueba_resultado <- reactive({...})
```

Ejecuta diversas pruebas según la opción seleccionada:

- t de Student (`t.test`)
- ANOVA (`aov`)
- Wilcoxon, correlación (`cor.test`)
- Shapiro-Wilk, Jarque-Bera
- Chi-cuadrado, McNemar, Kolmogorov-Smirnov

```
if (inherits(res, "htest")) {  
  p <- res$p.value  
  if (p < 0.05) {  
    cat(" p =", round(p, 4), "- Existe diferencia significativa.")  
  } else {  
    cat(" p =", round(p, 4), "- No hay diferencia significativa.")  
  }  
}
```

### 2.4.1. Interpretación Automática

Utiliza el paquete `report` para mostrar resultados interpretados en lenguaje natural.

## 2.5. Simulación del Teorema del Límite Central

```
observeEvent(input$simular_tlc, {  
  medias <- replicate(input$numero_muestras, {  
    mean(sample(variable, size = input$tamano_muestra, replace = TRUE))  
  })  
})
```

Grafica la distribución de medias para ilustrar la convergencia a la normalidad.

### 3. Resumen de Funcionalidades

Funcionalidad	Código Principal
Subir datos	<code>fileInput()</code> , <code>datos()</code>
Vista previa	<code>renderTable(head(...))</code>
Estadísticas descriptivas	<code>renderUI()</code> , <code>renderTable()</code>
Gráficos	<code>renderPlot()</code> con <code>ggplot2</code>
Pruebas estadísticas	<code>prueba_resultado()</code> , <code>renderPrint()</code>
Interpretación automática	<code>report::report()</code>
Simulación del TLC	<code>replicate()</code> , <code>renderPlot()</code>

### 4. Conclusión

Esta aplicación constituye una herramienta integral de análisis estadístico para usuarios con o sin experiencia en programación. Brinda soporte para cargar datos, explorar visual y numéricamente la información, realizar pruebas estadísticas relevantes y comprender los principios estadísticos mediante simulaciones.