

Ciencia de Datos II - Practica U1

Dashboard Interactivo de Análisis de
Tráfico Vehicular Peajes de Perú - R shiny

ALEX RAUL CRUZ ACCARAPI 26 de noviembre de 2025

Índice

1. RESUMEN	4
2. TECNOLOGÍAS Y HERRAMIENTAS	4
2.1. Lenguaje de Programación	4
2.2. Librerías Principales	4
2.3. Comandos de Instalación	4
3. ESTRUCTURA DEL PROYECTO	5
3.1. Arquitectura de Archivos	5
3.2. Arquitectura del Dashboard	5
4. VARIABLES DEL DATASET	5
4.1. Variables de Identificación	5
4.2. Variables Temporales	6
4.3. Variables de Tráfico Vehicular	6
4.3.1. Totales	6
4.3.2. Vehículos Ligeros	6
4.3.3. Vehículos Pesados por Ejes	6
5. FUNCIONALIDADES IMPLEMENTADAS	6
5.1. Panel de Control (Sidebar)	6
5.2. KPIs Principales	7
5.3. Visualizaciones Principales	7
5.3.1. Gráfico de Tráfico Mensual	7
5.3.2. Top 10 Peajes	7
5.3.3. Distribución Vehicular	7
5.3.4. Composición de Vehículos	7
5.3.5. Mapa de Peajes	8
5.3.6. Boxplot Comparativo	8
5.3.7. Composición Vehicular Detallada	8
6. ANÁLISIS AVANZADO	8
6.1. Clustering (Agrupación)	8
6.2. Análisis de Correlación	9
7. MODELOS PREDICTIVOS	9
7.1. ARIMA (AutoRegressive Integrated Moving Average)	9
7.2. Random Forest	10
7.3. Prophet (Facebook)	10
8. FUNCIONALIDADES DE EXPORTACIÓN	10
8.1. Exportación Simple	10
8.2. Reporte Completo Excel	11
9. DISEÑO Y EXPERIENCIA DE USUARIO	11
9.1. Sistema de Diseño	11
9.2. Componentes CSS	11

10.FLUJO DE DATOS	12
11.INSTRUCCIONES DE USO	12
11.1. Instalación	12
11.2. Ejecución	12
11.3. Interacción	13
12.VENTAJAS Y CARACTERÍSTICAS DESTACADAS	13
13.POSIBLES MEJORAS FUTURAS	14
14.CONCLUSIONES	14
15.REFERENCIAS Y RECURSOS	14
16.anexos	15

1. RESUMEN

El presente proyecto implementa un dashboard interactivo para el análisis y visualización de datos de tráfico vehicular en peajes del Perú. La aplicación permite explorar patrones de tráfico, realizar análisis avanzados mediante clustering y machine learning, y generar predicciones sobre el comportamiento futuro del tráfico vehicular.

Características Principales

- **Tecnología principal:** R Shiny
- **Tipo de aplicación:** Dashboard web interactivo
- **Fuente de datos:** CSV con 23 variables de tráfico vehicular
- **Análisis:** Clustering, predicciones y correlaciones

2. TECNOLOGÍAS Y HERRAMIENTAS

2.1. Lenguaje de Programación

- **R versión:** 4.x o superior
- **IDE recomendado:** RStudio

2.2. Librerías Principales

Librería	Versión	Propósito
shiny	-	Framework para aplicaciones web interactivas
dplyr	-	Manipulación y transformación de datos
ggplot2	-	Creación de gráficos estadísticos
plotly	-	Gráficos interactivos y visualizaciones dinámicas
DT	-	Tablas de datos interactivas
forecast	-	Modelos de series temporales (ARIMA)
randomForest	-	Algoritmos de machine learning
prophet	-	Predicciones de series temporales (Facebook)
writexl	-	Exportación de datos a Excel

Cuadro 1: Librerías utilizadas en el proyecto

2.3. Comandos de Instalación

```

1 # Instalar todas las librerías necesarias
2 install.packages(c(
3   "shiny", "dplyr", "ggplot2", "plotly", "DT",
4   "forecast", "randomForest", "prophet", "writexl"
5 ))

```

Listing 1: Instalación de librerías necesarias

3. ESTRUCTURA DEL PROYECTO

3.1. Arquitectura de Archivos

```

1 proyecto/
2 |
3 |-- app.R           # Archivo principal de ejecucion
4 |-- ui.R            # Interfaz de usuario (Frontend)
5 |-- server.R        # Logica del servidor (Backend)
6 '-- datos_limpios.csv # Dataset de entrada

```

Listing 2: Estructura de directorios

3.2. Arquitectura del Dashboard

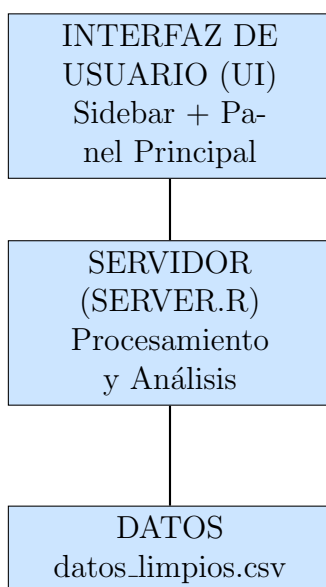


Figura 1: Arquitectura general del dashboard

4. VARIABLES DEL DATASET

4.1. Variables de Identificación

- CODIGO_PEAJE: Identificador único del peaje
- NOMBRE_PEAJE: Nombre descriptivo del peaje
- DEPARTAMENTO: Ubicación geográfica
- ADMINIST: Tipo de administración (Concesionado/No concesionado)

4.2. Variables Temporales

- ANIO: Año del registro
- MES: Mes del registro (1-12)

4.3. Variables de Tráfico Vehicular

4.3.1. Totales

- VEH_TOTAL: Total de vehículos
- VEH_LIGEROS_TOTAL: Total de vehículos ligeros
- VEH_PESADOS_TOTAL: Total de vehículos pesados

4.3.2. Vehículos Ligeros

- VEH_LIGEROS_AUTOMOVILES: Automóviles
- VEH_LIGEROS_TAR_DIF: Tarifa diferencial ligeros

4.3.3. Vehículos Pesados por Ejes

- VEH_PESADOS_TAR_DIF: Tarifa diferencial pesados
- VEH_PESADOS_2E a VEH_PESADOS_7E: Vehículos de 2 a 7 ejes

5. FUNCIONALIDADES IMPLEMENTADAS

5.1. Panel de Control (Sidebar)

Filtros Interactivos Disponibles

- Selección de departamento (Todos/Específico)
- Selección de administración (Todos/Concesionario/No concesionario)
- Selección de año
- Mapa geográfico interactivo con visualización de tráfico

Tecnologías utilizadas:

- `selectInput()`: Controles de selección
- `actionButton()`: Botón de aplicación de filtros
- `eventReactive()`: Reactividad controlada por eventos

5.2. KPIs Principales

Los indicadores clave de desempeño (KPIs) se calculan dinámicamente:

1. Total de vehículos en el período
2. Promedio de vehículos pesados
3. Número de peajes analizados
4. Administración predominante

Funciones clave:

```

1 sum()           # Agregacion de totales
2 mean()          # Calculo de promedios
3 length(unique()) # Conteo de valores unicos
4 table()          # Tablas de frecuencia

```

Listing 3: Cálculo de KPIs

5.3. Visualizaciones Principales

5.3.1. Gráfico de Tráfico Mensual

- **Tipo:** Líneas interactivas
- **Librería:** plotly
- **Función:** `plot_ly(type = 'scatter', mode = 'lines+markers')`
- **Agregación:** `group_by() + summarise()`

5.3.2. Top 10 Peajes

- **Tipo:** Lista ordenada
- **Función:** `arrange(desc()) + head(10)`
- **Renderizado:** `renderUI() + tags$ol()`

5.3.3. Distribución Vehicular

- **Tipo:** Gráfico de pie (circular)
- **Librería:** plotly
- **Función:** `plot_ly(type = 'pie')`

5.3.4. Composición de Vehículos

- **Tipo:** Barras horizontales
- **Propósito:** Desglose por tipo (ligeros/pesados)
- **Interactividad:** Hover con detalles

5.3.5. Mapa de Peajes

- **Tipo:** Barras horizontales dinámicas
- **Comportamiento:**
 - "Todos" → Muestra departamentos
 - Específico → Muestra peajes del departamento
- **Función:** Lógica condicional if/else

5.3.6. Boxplot Comparativo

- **Tipo:** Diagramas de caja
- **Análisis:** Distribución y valores atípicos
- **Función:** `plot_ly(type = 'box')`

5.3.7. Composición Vehicular Detallada

- **Tipo:** Barras apiladas horizontales
- **Variables:** 9 categorías de vehículos
- **Función:** `add_trace()` en bucle
- **Layout:** `barmode = 'stack'`

6. ANÁLISIS AVANZADO

6.1. Clustering (Agrupación)

Algoritmo: K-Means

Función principal: `kmeans()`

Variables de Entrada

- Total de vehículos
- Promedio de vehículos ligeros
- Promedio de vehículos pesados

Preprocesamiento:

```

1 scale()           # Normalizacion de datos
2 group_by()        # Agregacion por peaje
3 summarise()       # Calculo de estadisticas

```

Listing 4: Normalización de datos para clustering

Parámetros configurables:

- Número de clusters: 2-10

- Semilla aleatoria: `set.seed(42)`
- Inicializaciones: `nstart = 25`

Visualización:

- Gráfico de dispersión 2D
- Colores por cluster
- Información en hover

6.2. Análisis de Correlación

Matriz de correlación:

```
1 cor()           # Cálculo de correlaciones
2 na.omit()       # Manejo de valores faltantes
```

Listing 5: Cálculo de correlaciones

Visualización: Mapa de calor (heatmap)

Función: `plot_ly(type = "heatmap")`

Escala de colores: Rojo (-1) → Blanco (0) → Azul (+1)

7. MODELOS PREDICTIVOS

7.1. ARIMA (AutoRegressive Integrated Moving Average)

Librería: forecast

Función principal: `auto.arima()`

Proceso:

1. Conversión a serie temporal: `ts()`
2. Detección automática de parámetros
3. Generación de predicciones: `forecast()`
4. Intervalos de confianza al 95 %

Métricas de evaluación:

- **RMSE** (Root Mean Squared Error):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (1)$$

- **MAE** (Mean Absolute Error):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2)$$

- **R²** (Explained Variance)

7.2. Random Forest

Librería: randomForest

Función principal: randomForest()

Features engineered:

- lag1: Valor del mes anterior
- lag2: Valor de hace 2 meses
- trend: Tendencia temporal
- MES: Estacionalidad

Hiperparámetros:

- Número de árboles: 100
- Predicciones iterativas para horizonte futuro

7.3. Prophet (Facebook)

Librería: prophet

Función principal: prophet()

Ventajas de Prophet

- Manejo automático de estacionalidad
- Robusto a datos faltantes
- Intervalos de predicción automáticos

Configuración:

```
1 yearly.seasonality = TRUE
2 weekly.seasonality = FALSE
```

Listing 6: Configuración de Prophet

División de datos:

- Training: 80 %
- Testing: 20 %

8. FUNCIONALIDADES DE EXPORTACIÓN

8.1. Exportación Simple

- CSV de datos filtrados: write.csv()
- CSV de predicciones: downloadHandler()
- CSV de clusters: Resultados de agrupación

8.2. Reporte Completo Excel

Función: `write_xlsx()`

Hojas del reporte:

1. **Datos_Filtrados:** Dataset completo
2. **Resumen:** Estadísticas agregadas
3. **Top_Peajes:** Top 20 por tráfico
4. **Clusters:** Asignación de grupos
5. **Predicciones:** Valores futuros
6. **Metricas_Modelo:** Evaluación del modelo

9. DISEÑO Y EXPERIENCIA DE USUARIO

9.1. Sistema de Diseño

Paleta de colores:

Color	Código Hex
Azul primario	#007bff
Verde	#28a745
Rojo	#dc3545
Cian	#17a2b8
Gris	#6c757d

Cuadro 2: Paleta de colores del dashboard

Tipografía:

- Familia: Inter, -apple-system, sans-serif
- KPIs: 2rem, peso 700
- Subtítulos: 0.875rem

9.2. Componentes CSS

Cards (Tarjetas):

```

1 border-radius: 12px
2 box-shadow: 0 2px 8px rgba(0,0,0,0.1)
3 background: white

```

Listing 7: Estilo de tarjetas

Botones:

- Primarios: Azul con hover

- Secundarios: Gris
- Bordes redondeados: 6px

Responsividad:

- Sistema de grid de Shiny
- Márgenes adaptativos

10. FLUJO DE DATOS

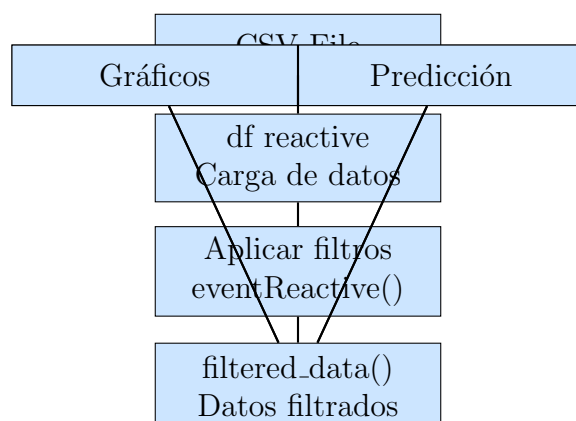


Figura 2: Flujo de datos en la aplicación

11. INSTRUCCIONES DE USO

11.1. Instalación

```

1 # 1. Instalar librerías
2 install.packages(c("shiny", "dplyr", "ggplot2", "plotly",
3                   "DT", "forecast", "randomForest",
4                   "prophet", "writexl"))
5
6 # 2. Cargar archivos del proyecto
7 # - app.R
8 # - ui.R
9 # - server.R
10 # - datos_limpios.csv
  
```

Listing 8: Pasos de instalación

11.2. Ejecución

```
1 # Opcion 1: Desde RStudio
2 # Abrir app.R y presionar "Run App"
3
4 # Opcion 2: Desde consola
5 library(shiny)
6 runApp("ruta/del/proyecto")
```

Listing 9: Ejecución de la aplicación

11.3. Interacción

1. **Seleccionar filtros** en el panel lateral
2. **Presionar "Aplicar filtros"** para actualizar visualizaciones
3. **Explorar gráficos** con interactividad (hover, zoom, pan)
4. **Generar clustering** con botón "Ejecutar Clustering"
5. **Crear predicciones** con botón "Generar predicción"
6. **Exportar resultados** desde la sección de descargas

12. VENTAJAS Y CARACTERÍSTICAS DESTACADAS

Características Principales

Interactividad completa: Todos los gráficos son interactivos

Análisis multinivel: Nacional → Departamental → Peaje individual

Machine Learning integrado: Clustering y predicciones

Diseño moderno: UI/UX profesional

Exportación completa: Múltiples formatos de salida

Responsive: Adaptable a diferentes tamaños de pantalla

Reactividad eficiente: Actualización controlada por eventos

Validación de datos: Manejo robusto de valores faltantes

13. POSIBLES MEJORAS FUTURAS

Roadmap de Desarrollo

- Integración con base de datos SQL
- Actualización automática de datos
- Autenticación de usuarios
- Más modelos predictivos (LSTM, XGBoost)
- Análisis de series temporales multivariadas
- Dashboard móvil nativo
- API REST para integración externa
- Notificaciones automáticas de alertas

14. CONCLUSIONES

El dashboard desarrollado constituye una herramienta integral para el análisis de tráfico vehicular, combinando visualización interactiva, análisis estadístico avanzado y capacidades predictivas. La arquitectura modular permite fácil mantenimiento y escalabilidad futura.

Tecnologías clave:

- R Shiny para la aplicación web
- Plotly para visualizaciones interactivas
- Algoritmos de ML para análisis avanzado
- Sistema de diseño coherente y profesional

El proyecto demuestra la capacidad de R para crear aplicaciones web completas y funcionales para análisis de datos, integrando múltiples técnicas de ciencia de datos en una interfaz intuitiva y profesional.

15. REFERENCIAS Y RECURSOS

- plataforma nacional de datos abiertos: <https://datosabiertos.gob.pe/dataset/flujo-vehicular-registrado-en-las-unidades-de-peaje-de-la-red-vial-nacional-2014-E2%80%932025-ii-0>
- Shiny Documentation: <https://shiny.rstudio.com/>
- Plotly R: <https://plotly.com/r/>



Figura 3: Un ejemplo de imagen insertada en LaTeX.



Figura 4: Un ejemplo de imagen insertada en LaTeX.

16. anexos

Fecha de elaboración: 26 de noviembre de 2025
Autor: ALEX RAUL CRUZ ACCARAPI
Institución: INGENIERIA ESTADISTICA E INFORMATICA

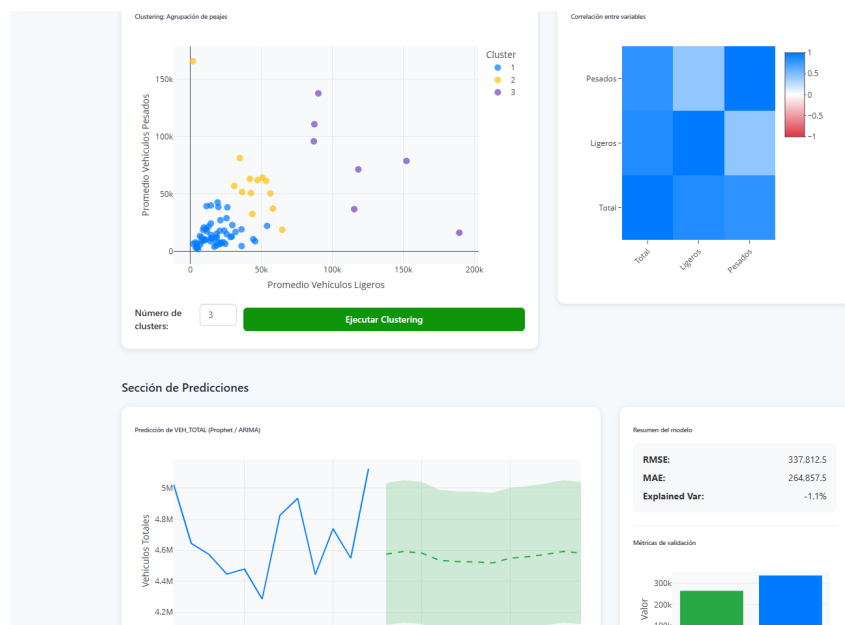


Figura 5: Un ejemplo de imagen insertada en LaTeX.