

# DASHBOARD INTERACTIVO DE HIDROCARBUROS

## **Análisis de Importación y Consumo en Guatemala**

### **Laboratorio 11 - Visualización Interactiva**

CC3066 - Data Science  
Universidad del Valle de Guatemala

Fecha: 30 De Octubre de 2025

#### **Autores:**

Diego Linares - #221256  
José Prince - #22087

## TABLA DE CONTENIDO

1.	Resumen Ejecutivo	3
2.	Selección de Paleta de Colores	3
3.	Planificación del Proyecto	4
4.	Diseño del Dashboard	5
5.	Selección de Herramienta	6
6.	Arquitectura del Sistema	7
7.	Modelos Predictivos	8
8.	Visualizaciones Implementadas	9
9.	Características de Interactividad	10
10.	Instrucciones de Uso	11

# 1. RESUMEN EJECUTIVO

Este proyecto presenta un dashboard interactivo desarrollado con **Streamlit** para el análisis exploratorio y predictivo de datos de importación y consumo de hidrocarburos en Guatemala. El sistema integra visualizaciones dinámicas enlazadas con tres modelos predictivos (Linear Regression, Random Forest y SARIMA) para facilitar la toma de decisiones basadas en datos.

El dashboard cumple con todos los requisitos establecidos en el laboratorio, incluyendo más de 8 visualizaciones interactivas, gráficos enlazados, comparación de modelos predictivos, y una interfaz intuitiva diseñada bajo principios de HCI y UX.

## 2. SELECCIÓN DE PALETA DE COLORES

### 2.1 Paleta Seleccionada

Color	Código Hex	Uso
Azul Principal	#1f77b4	Importación, gráficos principales
Naranja	#ff7f0e	Consumo, acciones
Verde	#2ca02c	Predicciones positivas, éxito
Rojo	#d62728	Alertas, errores, valores críticos
Azul Medio	#4a90e2	Gasolina Regular
Azul-Violeta	#7b68ee	Gasolina Superior
Naranja Cálido	#ff8c42	Diesel

### 2.2 Justificación

La paleta fue seleccionada siguiendo principios de diseño para dashboards corporativos:

- **Azul (#1f77b4):** Color principal que transmite confianza, profesionalismo y estabilidad. Ideal para datos de importación y análisis serio.
- **Naranja (#ff7f0e):** Color complementario que aporta energía y calidez, perfecto para representar consumo y acción.
- **Verde (#2ca02c):** Para indicadores positivos y predicciones favorables, asociado con crecimiento y éxito.
- **Rojo (#d62728):** Reservado para alertas, errores o valores críticos que requieren atención.
- **Grisés neutros:** Facilitan la lectura y reducen la fatiga visual, permitiendo que los datos sean el foco principal.

■ **Esta paleta cumple con:**

- Estándares de accesibilidad WCAG 2.1 para contraste
- Compatibilidad con daltonismo (deuteranopía y protanopía)
- Principios de UX para reducir fatiga visual

### 3. PLANIFICACIÓN DEL PROYECTO

#### 3.1 Distribución de Tareas

Tarea	Responsable	Tiempo
Diseño de arquitectura	Equipo	2 horas
Módulo de carga de datos	Diego Linares	3 horas
Módulo de modelos predictivos	José Prince	4 horas
Módulo de visualizaciones	Diego Linares	3 horas
Dashboard principal (Streamlit)	José Prince	5 horas
Integración y pruebas	Equipo	3 horas
Documentación y reporte	Equipo	2 horas

#### 3.2 Metodología

Se siguió una metodología ágil iterativa con los siguientes sprints:

- **Sprint 1:** Análisis de requisitos y diseño de arquitectura
- **Sprint 2:** Desarrollo de módulos base (carga de datos, modelos)
- **Sprint 3:** Desarrollo del dashboard y visualizaciones
- **Sprint 4:** Integración, pruebas y refinamiento de UX
- **Sprint 5:** Documentación y preparación de entrega

## 4. DISEÑO DEL DASHBOARD

### 4.1 Bosquejo de Diseño

El dashboard fue diseñado con una estructura de tres vistas principales:

- **Vista 1 - Exploración de Datos:** Permite al usuario explorar las series temporales, agregaciones anuales/mensuales, estadísticas descriptivas y distribuciones.
- **Vista 2 - Modelos Predictivos:** Muestra las predicciones de los tres modelos con métricas de desempeño y gráficos de ajuste.
- **Vista 3 - Comparación de Modelos:** Tabla y gráficos comparativos del desempeño de los modelos para facilitar la selección del mejor.

### 4.2 Estructura de Navegación

La navegación se implementó mediante un sidebar con radio buttons para cambiar entre vistas. Dentro de cada vista, se utilizan tabs y selectboxes para facilitar la exploración sin saturar la interfaz.

### 4.3 Principios de Diseño Aplicados

- **Jerarquía Visual:** Tamaños de fuente y colores diferenciados para guiar al usuario
- **Espaciado Generoso:** Uso de espacios en blanco para mejorar legibilidad
- **Consistencia:** Colores y estilos consistentes en todo el dashboard
- **Feedback Inmediato:** Indicadores de carga y tooltips informativos
- **Minimizar Carga Cognitiva:** No más de 3-4 controles visibles simultáneamente

## 5. SELECCIÓN DE HERRAMIENTA

### 5.1 Herramienta Seleccionada: Streamlit

Se seleccionó **Streamlit** como framework principal para el desarrollo del dashboard por las siguientes razones:

- **Facilidad de Uso:** Sintaxis simple en Python puro, sin necesidad de HTML/CSS/JavaScript
- **Desarrollo Rápido:** Permite crear prototipos funcionales en horas
- **Integración con Plotly:** Soporte nativo para gráficos interactivos
- **Despliegue Gratuito:** Streamlit Cloud permite publicar gratuitamente
- **Comunidad Activa:** Amplia documentación y ejemplos disponibles
- **Compatibilidad:** Funciona con todas las librerías de Python (pandas, scikit-learn, etc.)

### 5.2 Librerías Complementarias

Librería	Versión	Propósito
streamlit	1.31.0	Framework del dashboard
plotly	5.18.0	Visualizaciones interactivas
pandas	2.1.4	Manipulación de datos
scikit-learn	1.4.0	Modelos de ML
statsmodels	0.14.1	Modelo SARIMA
numpy	1.26.3	Operaciones numéricas

## 6. ARQUITECTURA DEL SISTEMA

### 6.1 Estructura Modular

El proyecto sigue una arquitectura modular para facilitar mantenimiento y escalabilidad:

- **config.py**: Configuración central (colores, parámetros, textos)
- **utils/data\_loader.py**: Carga y preprocesamiento de datos
- **utils/predictive\_models.py**: Entrenamiento y evaluación de modelos
- **utils/visualization\_utils.py**: Funciones para crear gráficos con Plotly
- **app.py**: Aplicación principal de Streamlit

### 6.2 Flujo de Datos

1. **Carga de Datos**: DataLoader lee archivos Excel y preprocesa
2. **Procesamiento**: Se calculan agregaciones y estadísticas
3. **Modelado**: PredictiveModels entrena y evalúa modelos
4. **Visualización**: Funciones crean gráficos interactivos
5. **Presentación**: Streamlit renderiza en el navegador

### 6.3 Optimizaciones Implementadas

- **Caching**: @st.cache\_data para evitar recarga de datos en cada interacción
- **Carga Lazy**: Los modelos se entrenan solo cuando el usuario los requiere
- **Normalización**: Datos normalizados para mejorar desempeño de modelos
- **Vectorización**: Uso de operaciones vectorizadas de NumPy/Pandas



## 7. MODELOS PREDICTIVOS IMPLEMENTADOS

### 7.1 Linear Regression

**Descripción:** Modelo de regresión lineal simple que utiliza 12 períodos anteriores como características (lookback=12).

**Ventajas:** Simple, rápido, fácil de interpretar, bajo riesgo de overfitting.

**Limitaciones:** Asume relaciones lineales, no captura patrones complejos.

### 7.2 Random Forest

**Descripción:** Modelo de ensamble basado en árboles de decisión con 100 árboles y profundidad máxima de 10.

**Ventajas:** Captura relaciones no lineales, robusto a outliers, reduce varianza.

**Limitaciones:** Más lento que regresión lineal, menos interpretable.

### 7.3 SARIMA (Seasonal ARIMA)

**Descripción:** Modelo estadístico especializado para series temporales con estacionalidad. Orden (1,1,1) y estacionalidad (1,1,1,12).

**Ventajas:** Diseñado específicamente para series de tiempo, captura estacionalidad.

**Limitaciones:** Computacionalmente costoso, requiere datos estacionarios.

### 7.4 Métricas de Evaluación

- **MAE (Mean Absolute Error):** Promedio de errores absolutos. Fácil de interpretar.
- **MSE (Mean Squared Error):** Promedio de errores al cuadrado. Penaliza más errores grandes.
- **R<sup>2</sup> (Coefficient of Determination):** Proporción de varianza explicada (0-1).
- **RMSE (Root Mean Squared Error):** Raíz del MSE, en unidades originales.

## 8. VISUALIZACIONES IMPLEMENTADAS

El dashboard incluye más de 8 visualizaciones interactivas:

#	Visualización	Descripción	Interactividad
1	Series Temporales	Evolución de importación/consumo	Zoom, pan, hover
2	Comparación Import/Consumo	Gráfico de área superpuesto	Filtros enlazados
3	Volumen Anual	Barras agrupadas por año	Hover tooltips
4	Patrones Mensuales	Líneas de promedios mensuales	Zoom, pan
5	Matriz Correlación	Heatmap de correlaciones	Hover valores
6	Distribuciones	Histograma + boxplot	Selector productos
7	Predicciones	Real vs predicho	Selector modelos
8	Scatter Real/Pred	Análisis de ajuste	Hover, zoom
9	Métricas Comparativas	Barras de métricas	Selector modelos
10	Estadísticas Tabla	DataFrame interactivo	Sort, filtros

## 9. CARACTERÍSTICAS DE INTERACTIVIDAD

### 9.1 Gráficos Enlazados

El dashboard implementa múltiples niveles de enlace entre visualizaciones:

- **Enlace 1:** Al seleccionar un producto en series temporales, el gráfico de todos los productos se actualiza para destacar el seleccionado.
- **Enlace 2:** Al cambiar el tipo de dato (Importación/Consumo), todas las visualizaciones de la sección se actualizan automáticamente.
- **Enlace 3:** En comparación de modelos, la selección de modelos afecta tanto la tabla como los gráficos de métricas y predicciones.

### 9.2 Niveles de Detalle

Los usuarios pueden aumentar o disminuir el nivel de detalle mediante:

- Selección de productos específicos vs todos los productos
- Vista agregada anual vs vista mensual detallada
- Estadísticas resumidas vs distribuciones completas
- Métricas agregadas vs predicciones punto por punto

### 9.3 Controles de Usuario

- **Selectbox:** Selección única de opciones (productos, modelos)
- **Multiselect:** Selección múltiple para comparaciones
- **Radio Buttons:** Cambio entre vistas principales
- **Tabs:** Organización de contenido relacionado
- **Expanders:** Información adicional colapsable
- **Tooltips:** Ayuda contextual en gráficos

## 10. INSTRUCCIONES DE USO

### 10.1 Instalación

#### Opción 1: Script Automático

Linux/Mac:

```
./setup.sh
```

Windows:

```
setup.bat
```

#### Opción 2: Manual

1. Crear entorno virtual: `python -m venv venv`
2. Activar: `source venv/bin/activate` (Linux/Mac) o `venv\Scripts\activate` (Windows)
3. Instalar dependencias: `pip install -r requirements.txt`

### 10.2 Ejecución

Ejecutar desde la carpeta del proyecto:

```
streamlit run app.py
```

El dashboard se abrirá automáticamente en el navegador en <http://localhost:8501>

### 10.3 Navegación

- Usar el **sidebar** izquierdo para cambiar entre vistas
- Usar **selectboxes** para elegir productos y modelos
- Usar **tabs** para explorar diferentes análisis
- Hacer **hover** sobre gráficos para ver detalles
- Hacer **zoom** arrastrando sobre gráficos
- Hacer **doble clic** para resetear zoom

## CONCLUSIONES

1. Se desarrolló exitosamente un dashboard interactivo que cumple con todos los requisitos del laboratorio, incluyendo más de 8 visualizaciones, gráficos enlazados, y comparación de 3 modelos predictivos.
2. La selección de Streamlit como framework permitió un desarrollo ágil y una interfaz intuitiva que facilita la exploración de datos sin necesidad de conocimientos técnicos avanzados por parte del usuario final.
3. La paleta de colores seleccionada cumple con estándares de accesibilidad y mejora la legibilidad y usabilidad del dashboard, siguiendo principios de diseño corporativo.
4. Los tres modelos predictivos implementados (Linear Regression, Random Forest, SARIMA) ofrecen diferentes enfoques para el análisis de series temporales, permitiendo comparar su desempeño y seleccionar el más adecuado según el caso de uso.
5. La arquitectura modular del proyecto facilita el mantenimiento, pruebas y futuras extensiones del sistema.

## RECOMENDACIONES

1. Considerar el despliegue en Streamlit Cloud para acceso remoto sin instalación local.
2. Implementar autenticación si el dashboard será usado en producción con datos sensibles.
3. Añadir funcionalidad de exportación de gráficos y tablas para reportes ejecutivos.
4. Incorporar más modelos de ML avanzados (LSTM, Prophet) para series con patrones complejos.
5. Implementar sistema de alertas automáticas cuando se detecten anomalías en los datos.