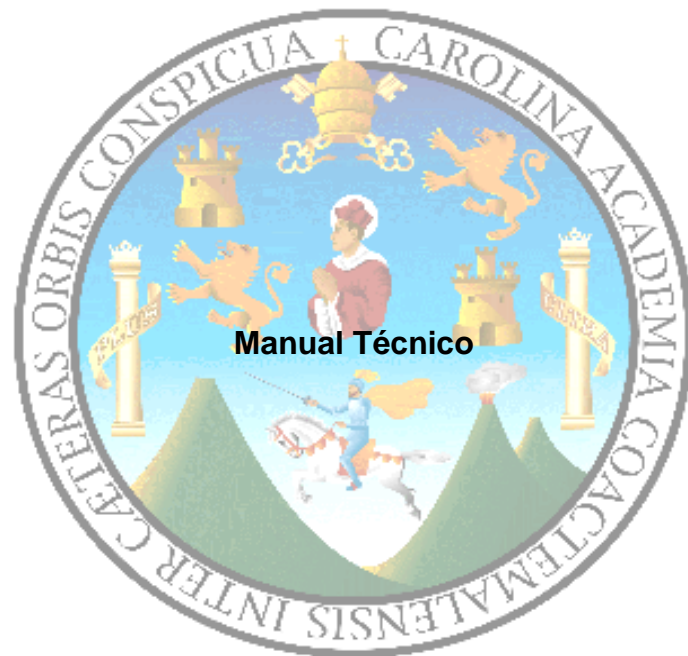


**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**  
**FACULTAD DE INGENIERIA**  
**ESCUELA DE CIENCIAS Y SISTEMAS**  
**INTELIGENCIA ARTIFICIAL 1**  
**ING. LUIS FERNADO ESPINO BARRIOS**  
**AUX: ERICK SANDOVAL**



**Manual Técnico**

<b>Carné</b>	<b>Nombre</b>
201602530	Andres Eduardo Pontaza Muralles

## índice

Objetivos .....	2
Sección de pasos para el uso de las tecnologías.....	3
Descripción de métodos utilizados .....	7
Descripción de los archivos CSV creados .....	8

## Objetivos

1. Implementar modelos de machine learning utilizando la librería titusjs.
2. Aprender a utilizar los diferentes modelos o algoritmos que posee la librería titusjs.
3. Desarrollar visualizaciones y resultados de los modelos.
4. Desplegar la app en Github Pages.
5. Uso de dataframe en JS.

## Sección de pasos para el uso de las tecnologías

### HTML

El HTML es la estructura principal del sitio. Define la disposición de secciones como la carga de datos, las secciones específicas para cada modelo con la visualización de los gráficos y los resultados.

```
<!-- Sección de carga de CSV y Previsualizaion de Datos -->
<section class="section">...
</section>

<!-- Sección de configuración de modelo -->
<section class="section">...
</section>

<!-- Secciones específicas de cada modelo -->

<!-- Predicción -->
<!-- Linear Regression -->
<div id="linearRegressionSection" class="model-section" style="display: none;">...
</div>

<!-- Tendencia -->
<!-- Polynomial Regression -->
<div id="polynomialRegressionSection" class="model-section" style="display: none;">
</div>

<!-- Patrones -->
<!-- Decision Tree -->
<section id="decisionTreeSection" class="section" style="display: none;">...
</section>
```

## JS

En JavaScript se tienen las funciones para manejar el entrenamiento de cada modelo. Incluye código para procesar el CSV, entrenar el modelo, y visualizar los diferentes gráficos.

Como se observa se carga el csv, luego se crea un dataframe del csv luego se va actualizando el contenido según el modelo seleccionado y posee la lógica de cada método o algoritmo con sus funciones para graficar.

```
// DataFrame global
let df;

// Funcion para cargar un archivo CSV y crear el DataFrame
function loadCSV() { ...
}

// Función para obtener los valores de las columnas seleccionadas
function fillColumnSelectors(columns) { ...
}

// Función para obtener los valores de las columnas seleccionadas
function fillColumnSelectorsPoly(columns) { ...
}

// Función para mostrar una previsualización de los datos cargados
function displayDataPreview(data) { ...
}

// Función para actualizar el nombre del modelo seleccionado
function updateModelName() { ...
}

// Función para actualizar los parámetros del modelo seleccionado
function updateModelName() { ...
}

// Función de entrenamiento para Linear Regression
function trainLinearRegression() { ...
}

// Función para entrenar y predecir con Polynomial Regression
function trainPolynomialRegression() { ...
}

// Función de entrenamiento para Decision Tree
function trainDecisionTree() { ...
}
```

## CSS

Usa CSS para estilizar la página, mejorando la experiencia de usuario. Define estilos para las secciones, los botones, y los resultados.

```
/* Estilo general */
* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
  font-family: Arial, sans-serif;
}

body {
  background: #16222A;
  background: -webkit-linear-gradient(to right, #3A6073, #16222A);
  background: linear-gradient(to right, #3A6073, #16222A);
}

/* Contenedor principal */
.container {
  max-width: 1200px;
  margin: 20px auto;
  padding: 20px;
  background-color: #434f57;
  border-radius: 10px;
  box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
  text-align: center;
}

/* Títulos */
h1 {
  color: #ffffff;
  font-size: 2em;
  margin-bottom: 20px;
}

h2 {
  color: #000000;
  font-size: 1.5em;
  margin: 10px 0;
}
```

## Librerías JS Puro

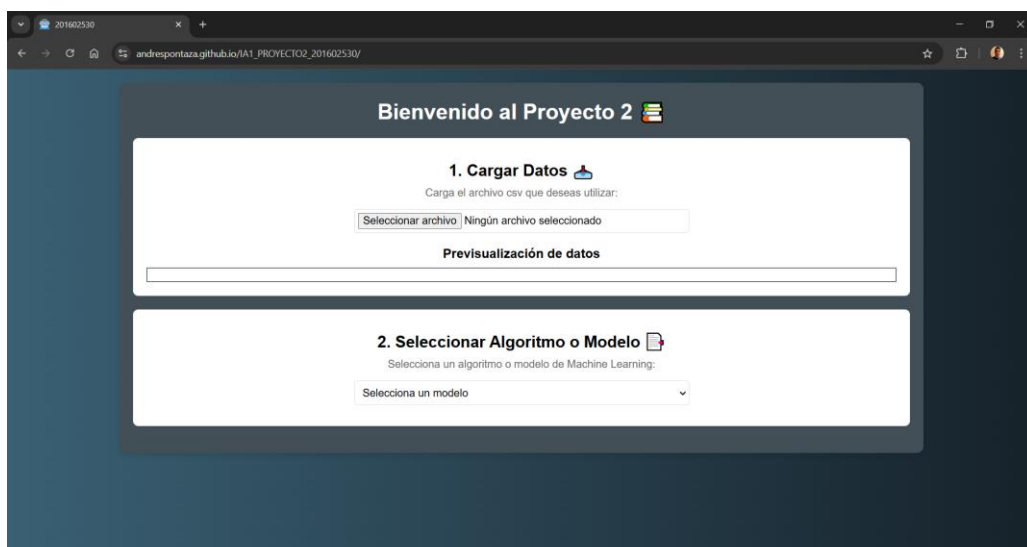
Para el proyecto se utilizaron librerías en JavaScript puro para el manejo de datos, visualización y machine learning.

1. **TytusJS** (tytus.js): Esta biblioteca te permite implementar algoritmos de Machine Learning directamente en JavaScript.
2. **PapaParse** (papaparse.min.js): Es una biblioteca diseñada para facilitar el manejo de archivos CSV. PapaParse permite leer y parsear archivos CSV de forma rápida y eficiente.
3. **DanfoJS** (bundle.min.js): Similar a Pandas en Python, DanfoJS proporciona funcionalidades para trabajar con DataFrames en JavaScript.
4. **Google Charts** (loader.js): Esta es una herramienta de visualización para mostrar gráficos.
5. **Vis.js** (vis-network.min.js): Vis.js permite crear y manipular gráficos de red.
6. **app.js**: Este archivo contiene el código JavaScript personalizado donde se integra todas las funcionalidades y lógica de la aplicación.

```
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>201602530</title>
<link rel="icon" type="image/x-icon" href="images/favicon.ico">
<link rel="stylesheet" href="style.css">
<script src="https://tytusdb.github.io/tytusjs/dist/tytus.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/PapaParse/5.3.0/papaparse.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/danfojs@1.1.2/lib/bundle.min.js"></script>
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/vis/4.21.0/vis-network.min.js"></script>
<script src="app.js"></script>
</head>
```

## Github Pages

GitHub Pages permite alojar el proyecto de manera gratuita y acceder a él desde cualquier lugar.



## Descripción de métodos utilizados

1. **Predicción (Linear Regression):** La **predicción** se refiere a la estimación de valores futuros de una variable dependiente (Y) a partir de una variable independiente (X) usando una línea recta.

```
// Función de entrenamiento para Linear Regression  
function trainLinearRegression() { ...  
}
```

2. **Tendencia (Polynomial Regression):** Se busca modelar la **tendencia** en los datos que no se ajustan bien a una línea recta, sino a una curva. Aquí, el modelo es una función polinómica que permite capturar variaciones más complejas y tendencias en los datos.

```
// Función para entrenar y predecir con Polynomial Regression  
function trainPolynomialRegression() { ...  
}
```

3. **Patrones (Decision Tree):** En un **Árbol de Decisión**, el modelo busca identificar **patrones** en los datos al dividir el conjunto de datos en subconjuntos basados en reglas lógicas. Este proceso de segmentación ayuda a entender qué características o combinaciones de valores de las variables explican mejor los resultados.

```
// Función de entrenamiento para Decision Tree  
function trainDecisionTree() { ...  
}
```



## Descripción de los archivos CSV creados

1. **Datos Lineal:** El archivo CSV con las columnas **X** e **Y** proporciona una base fundamental para entrenar y evaluar Linear Regression de Machine Learning en el proyecto.

```
datos_lineal.csv X
assets > datos_lineal.csv > data
1 X,Y
2 7,2
3 1,9
4 10,2
5 5,5
6 4,7
7 3,11
8 13,2
9 10,5
10 2,14
11 |
```

2. **Datos Poly:** Posee los vectores **xTrain**, **yTrain** y **predictArray** en el contexto del proyecto de Machine Learning. Estos vectores representan los datos de entrenamiento y los valores a predecir para Polynomial Regression.

```
datos_poly.csv X
assets > datos_poly.csv > data
1 xTrain,yTrain,predictArray
2 0,0,0
3 1,15,1
4 2,5,2
5 3,25,3
6 4,40,4
7 5,35,5
8 6,10,6
9 7,5,7
10 8,45,8
11 9,35,9|
```

3. Datos Tree: Representa distintas condiciones climáticas y su relación con la decisión de jugar tenis (columna "Play Tennis"). Los datos están organizados en cinco columnas, donde cada columna representa una característica o factor relevante. La información puede es utilizada, para encontrar un Patrón en el modelo de clasificación de un Árbol de Decisión

```
datos_tree.csv X
assets > datos_tree.csv > data
1 Outlook, Temperature, Humidity, Windy, Play Tennis
2 Sunny, Hot, High, Weak, No
3 Sunny, Hot, High, Strong, No
4 Overcast, Hot, High, Weak, Yes
5 Rainy, Mild, High, Weak, Yes
6 Rainy, Cool, Normal, Weak, Yes
7 Rainy, Cool, Normal, Strong, No
8 Overcast, Cool, Normal, Strong, Yes
9 Sunny, Mild, High, Weak, No
10 Sunny, Cool, Normal, Weak, Yes
11 Rainy, Mild, Normal, Weak, Yes
12 Sunny, Mild, Normal, Strong, Yes
13 Overcast, Mild, High, Strong, Yes
14 Overcast, Hot, Normal, Weak, Yes
15 Rainy, Mild, High, Strong, No
```