



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ

FACULTAD DE INGENIERÍA Y CIENCIAS COMPUTACIONALES

LICENCIATURA EN INGENIERÍA DE SOFTWARE

INGENIERÍA WEB

Ensayo: Mini Proyecto #2 – Sentencias de Control y Clases

Autor: Britney Valoy (8-998-213)

Grupo: 1SF132

Profesor: Ing. Irina Fong

Fecha de entrega: viernes, 7 de noviembre de 2025

Tabla de contenido

Introducción	3
Metodología	3
Funciones.....	4
1. Matemáticas.....	4
2. Cadenas	4
3. Validación de Datos	4
4. Clases y Métodos Utilitarios.....	5
Lecciones Aprendidas.....	6
Ejecución de Programa	7
Figura 1. Menú principal	7
Figura 2. Problema 1	7
Figura 3. Problema 2	8
Figura 4. Problema 3	8
Figura 5. Problema 4	9
Figura 6. Problema 5	9
Figura 7. Problema 6	10
Figura 8. Problema 7	10
Figura 9. Problema 8	11
Figura 10. Problema 9	11
Figura 11. Problema 10	12

Introducción

El presente informe resume el desarrollo del Mini Proyecto #2 de Ingeniería Web, cuyo propósito fue aplicar las estructuras de control, funciones y clases del lenguaje PHP para resolver diferentes problemas algorítmicos en un entorno web. El proyecto tuvo como objetivo fortalecer las competencias en el diseño de aplicaciones web funcionales siguiendo las prácticas de programación definidas por el estándar PSR-1. Para ello, se desarrollaron diez programas independientes, cada uno con un enfoque distinto, integrando formularios HTML, validación de datos y un menú principal que facilita la navegación entre las soluciones. Finalmente, el informe presenta los resultados obtenidos y las evidencias visuales que demuestran la correcta ejecución de los programas y el cumplimiento de los objetivos académicos planteados.

Metodología

El desarrollo del presente mini proyecto se llevó a cabo aplicando un enfoque práctico de programación estructurada y orientada a objetos en el lenguaje PHP. El objetivo principal fue resolver una serie de problemas algorítmicos utilizando sentencias de control, estructuras repetitivas, funciones y clases, siguiendo las prácticas de codificación establecidas por el estándar PSR. El proceso inició con la revisión de los enunciados de cada problema. Se identificaron los requerimientos específicos y se definió la lógica necesaria para su resolución. Posteriormente, se diseñaron las soluciones considerando los principios de modularidad y reutilización del código, mediante la creación de funciones y clases con métodos estáticos que hace uso de operaciones comunes, como cálculos matemáticos, validación de datos y generación de salidas.

Las entradas de usuario fueron validadas y sanitizadas utilizando funciones nativas como `filter_var()` y `htmlspecialchars()`, con el propósito de prevenir vulnerabilidades y garantizar la integridad de la información procesada. Para facilitar la navegación, se desarrolló un menú principal dinámico, desde el cual el usuario puede seleccionar el problema a ejecutar. De igual forma, se incorporó un enlace de retorno al menú principal. El diseño del código fuente consideró la claridad y estandarización de los nombres de variables y métodos aplicando las convenciones de nomenclatura en camelCase y StudlyCaps. Asimismo, se incluyó un footer externo que muestra la fecha actual de ejecución, para poder reforzar la separación lógica de los componentes y mejorar la mantenibilidad del sistema. Finalmente, la ejecución de cada programa fue probada y verificada para asegurar la corrección de los resultados. Se documentaron las soluciones y se recopilaron capturas de pantalla representativas de la salida de cada uno de los problemas como evidencia del funcionamiento correcto del código y del cumplimiento de los criterios de usabilidad y accesibilidad establecidos.

Funciones

1. Matemáticas

Las funciones matemáticas fueron importantes para la resolución de los distintos problemas del proyecto, particularmente aquellos relacionados con cálculos estadísticos y numéricos. Entre las operaciones más destacadas se incluyen:

- **Suma de números consecutivos:** se utilizaron estructuras *for* y *while* para calcular sumatorias, como en el problema de la suma de los números del 1 al 1,000.
- **Promedio y media aritmética:** utilizando *array_sum()* y *count()* para calcular el promedio de un conjunto de notas o valores.
- **Desviación estándar:** implementada mediante una función personalizada que calcula la varianza y obtiene la raíz cuadrada usando la función nativa *sqrt()*.
- **Potencias y multiplicaciones repetitivas:** se aplicó *pow()* para generar las primeras potencias de un número ingresado.
- **Valores mínimos y máximos:** se emplearon las funciones *min()* y *max()* para determinar los límites de un conjunto de datos.

2. Cadenas

Las funciones de manejo de cadenas se implementaron con el objetivo de limpiar, estandarizar y formatear los datos ingresados por el usuario. Esto permitió mejorar la presentación y garantizar la consistencia de la información procesada.

- *trim()* para eliminar espacios en blanco innecesarios en los extremos de las cadenas.
- *htmlspecialchars()* para codificar caracteres especiales y evitar la ejecución de código malicioso en las salidas HTML.
- *Concatenación de cadenas* para construir mensajes personalizados y estructurar salidas de texto.

Gracias a estas funciones, la aplicación mantiene una presentación limpia y previene errores ocasionados por formatos inconsistentes o entradas mal estructuradas.

3. Validación de Datos

La validación de datos fue una parte esencial del proyecto, ya que permitió garantizar la integridad de la información introducida por los usuarios a través de los formularios. Para ello se utilizaron las siguientes funciones y técnicas:

- *filter_var()*, empleada junto con filtros como *FILTER_VALIDATE_INT* y *FILTER_VALIDATE_FLOAT* para asegurar que los valores ingresados correspondan al tipo esperado.
- *htmlspecialchars()*, aplicada para codificar las entradas y proteger la aplicación frente a ataques de tipo Cross-Site Scripting (XSS).

Estas validaciones se implementaron tanto en las funciones individuales como en clases dedicadas al control de entrada para promover la reutilización del código y la seguridad general del sistema.

4. Clases y Métodos Utilitarios

Además de las funciones tradicionales, se diseñaron clases utilitarias con métodos estáticos que agrupan operaciones de uso frecuente. Estas clases permitieron centralizar tareas repetitivas, como cálculos estadísticos o validaciones siguiendo principios básicos de la Programación Orientada a Objetos (POO).

- **Calculos:** *MathUtils::mean()*
- **Validaciones:** *Validation::isNumeric()*
- **Utilidades:** *Navigation::backToMenu()*

La utilización de clases con métodos estáticos favoreció la legibilidad del código, la separación lógica por responsabilidades y el cumplimiento del estándar PSR-1, que establece convenciones de nomenclatura (StudlyCaps para clases y camelCase para métodos).

Lecciones Aprendidas

Durante la realización del mini proyecto se adquirieron conocimientos importantes relacionados con el desarrollo web en PHP, la estructura lógica de los algoritmos y la importancia de las buenas prácticas de programación. Entre las principales lecciones aprendidas se destacan las siguientes:

- **Comprensión de las estructuras de control:** la práctica permitió afianzar el uso de sentencias condicionales (if, switch) y estructuras repetitivas (for, while, foreach), comprendiendo cómo aplicarlas de manera eficiente según el tipo de problema.
- **Uso de funciones para modularizar el código:** se reconoció la importancia de dividir las tareas en funciones independientes, lo cual facilita la lectura, el mantenimiento y la detección de errores.
- **Importancia de la validación y sanitización de datos:** se evidenció la necesidad de proteger las aplicaciones frente a entradas no seguras, implementando funciones como *filter_var()*, *preg_match()* y *htmlspecialchars()* para fortalecer la seguridad.
- **Aplicación de principios de la programación orientada a objetos:** el uso de clases y métodos estáticos permitió comprender cómo organizar y reutilizar código de manera más profesional.
- **Integración entre frontend y backend:** al trabajar con formularios HTML conectados con scripts PHP, se comprendió la interacción entre el cliente y el servidor, así como el flujo completo de los datos.
- **Dificultades encontradas:** algunos problemas requerían cálculos complejos o manipulación de arreglos bidimensionales, lo que implicó investigar funciones nativas y depurar errores de lógica. También se presentaron desafíos al manejar las rutas entre archivos y formularios, los cuales fueron superados mediante la revisión cuidadosa del flujo de ejecución.

En conjunto, el proyecto fortaleció las competencias en el diseño de aplicaciones web funcionales, seguras y estructuradas. A través del proyecto se evidencia la utilidad de las sentencias de control, las estructuras repetitivas y las clases en el desarrollo de soluciones efectivas en PHP.

Ejecución de Programa

Figura 1. Menú principal del Mini Proyecto

Interfaz inicial del sistema donde se presentan los diez problemas del proyecto. Desde este menú el usuario puede acceder a cada uno de los ejercicios implementados en PHP.

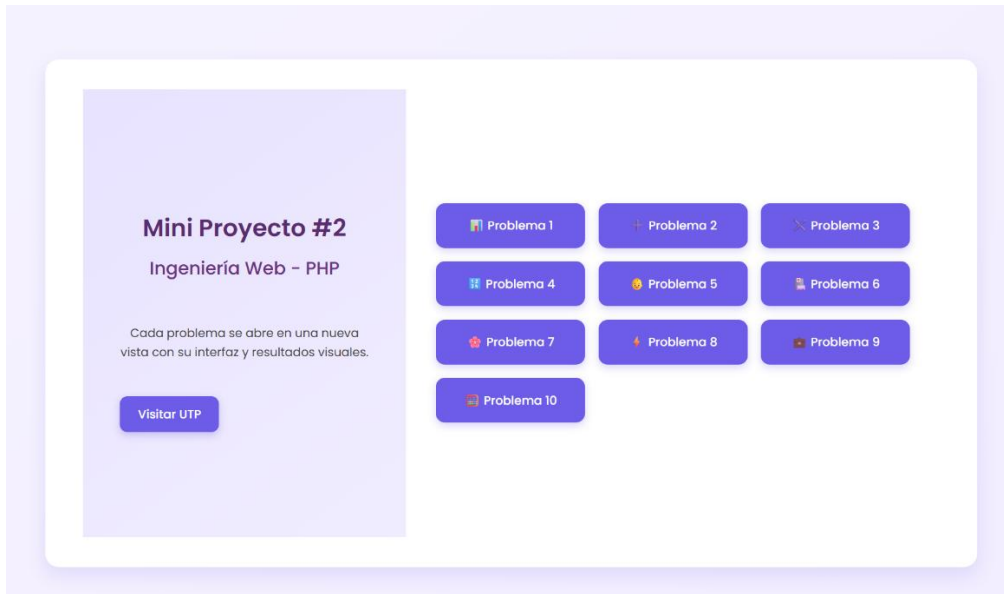


Figura 2. Problema 1 – Cálculo de media, desviación estándar, mínimo y máximo

Formulario que permite ingresar cinco números y calcular la media, desviación estándar, valor mínimo y máximo. Los resultados se presentan de forma ordenada en pantalla.

The image shows a web browser window displaying the 'Problema 1' form. The title is 'Problema 1' and the subtitle is 'Cálculo de Media, Desviación Estándar, Mínimo y Máximo'. There are five input fields labeled 'Número 1' through 'Número 5'. Below the input fields is a blue 'Calcular' button. Underneath the button is a box titled 'Resultados Estadísticos' containing four items: 'Media: 16.60', 'Desviación Estándar: 22.22', 'Valor Mínimo: 2', and 'Valor Máximo: 56'. At the bottom of the form is a link that says 'Volver al menú principal'. The browser's address bar shows 'localhost:8000/problems/problema1.php'.

Figura 3. Problema 2 – Suma de números del 1 hasta N

Programa que solicita un número N e imprime la suma de todos los números desde 1 hasta el valor ingresado, utilizando tanto un ciclo for como una fórmula matemática. Ambos métodos arrojan el mismo resultado.

The screenshot shows a web browser at localhost:8000/problems/problema2.php. The page title is 'Problema 2' with the subtitle 'Calcular la suma de los números del 1 hasta N'. It features an input field for 'Ingrese un número N:' with the value '1000' and a 'Calcular' button. Below this, a yellow box displays the results: 'Usando ciclo for: 500,500' and 'Usando fórmula matemática $n(n+1)/2$: 500,500'. It concludes with 'Ambos métodos arrojan el mismo resultado.' and a link 'Volver al menú principal'. The footer identifies the 'Universidad Tecnológica de Panamá - Ingeniería Web' and the execution date '05/11/2025'.

Figura 4. Problema 3 – Múltiplos de 4

Formulario que permite ingresar un número y visualizar los primeros N múltiplos de 4. El resultado se muestra en una lista estructurada generada mediante un ciclo "for".

The screenshot shows a web browser at localhost:8000/problems/problema3.php. The page title is 'Problema 3' with the subtitle 'Imprimir los N primeros múltiplos de 4'. It features an input field for 'Ingrese la cantidad de múltiplos:' with the value '5' and a 'Mostrar' button. Below this, the text 'Primeros 5 múltiplos de 4:' is followed by a list of five multiplication results: '4 x 1 = 4', '4 x 2 = 8', '4 x 3 = 12', '4 x 4 = 16', and '4 x 5 = 20'. A link 'Volver al menú principal' is at the bottom left. The footer identifies the 'Universidad Tecnológica de Panamá - Ingeniería Web' and the execution date '05/11/2025'.

Figura 5. Problema 4 – Suma de números pares e impares

Aplicación que calcula de manera independiente la suma de los números pares e impares comprendidos entre 1 y 200. Los resultados se presentan en pantalla junto a una gráfica comparativa.

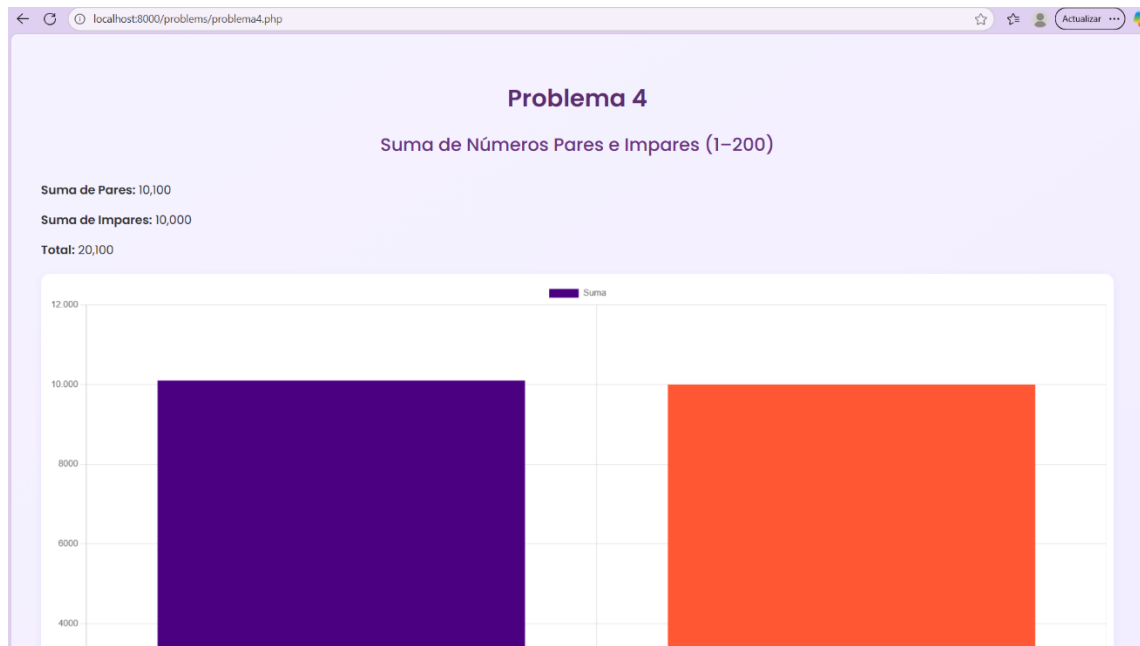


Figura 6. Problema 5 – Clasificación por edad

Programa que permite ingresar la edad de cinco personas y clasificarlas según su grupo etario (niño, adolescente, adulto o adulto mayor). Incluye una gráfica circular que muestra la distribución de las categorías.

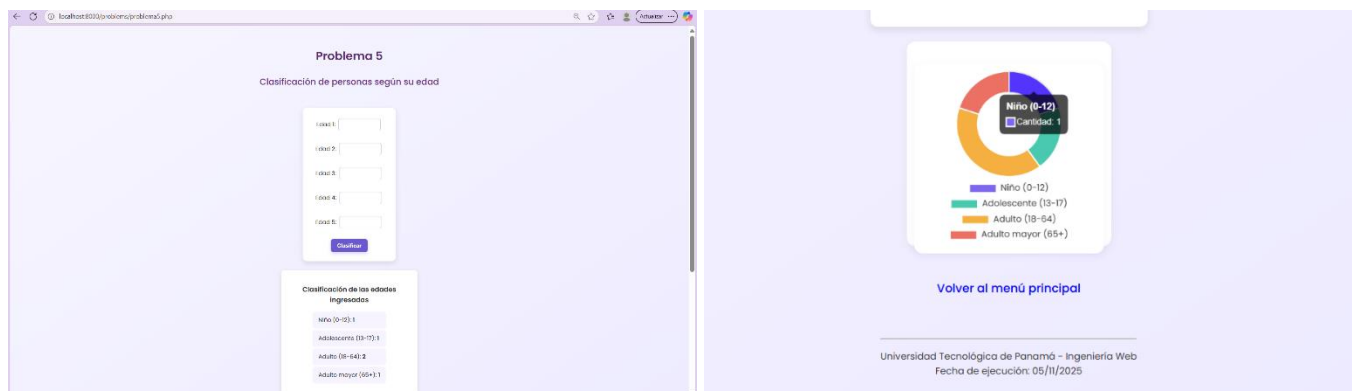


Figura 7. Problema 6 – Distribución del presupuesto hospitalario

Formulario que solicita el presupuesto total del hospital y calcula la distribución porcentual correspondiente a las áreas de Ginecología, Traumatología y Pediatría. Los resultados se muestran numéricamente y mediante un gráfico circular.

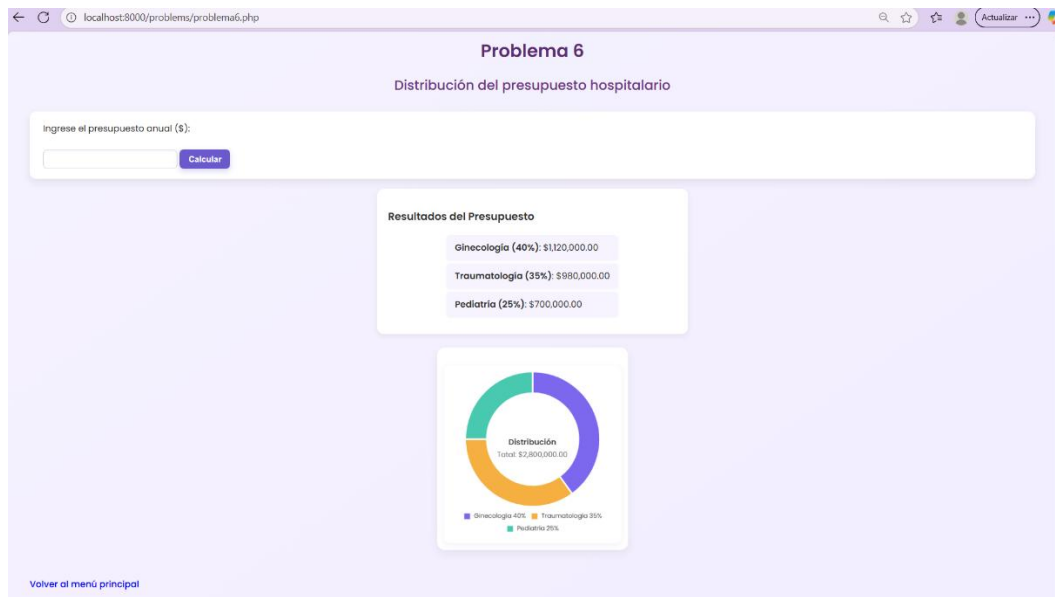


Figura 8. Problema 7 – Calculadora de promedio, desviación estándar, nota mínima y máxima

Aplicación que permite ingresar una cantidad de notas y, posteriormente, calcular valores estadísticos fundamentales, tales como promedio, desviación estándar y extremos de los datos. La interfaz incluye una representación gráfica de los resultados.



Figura 9. Problema 8 – Estación del año

Programa que determina la estación del año (Verano, Otoño, Invierno o Primavera) según la fecha ingresada por el usuario. Presenta una imagen ilustrativa asociada a la estación correspondiente.

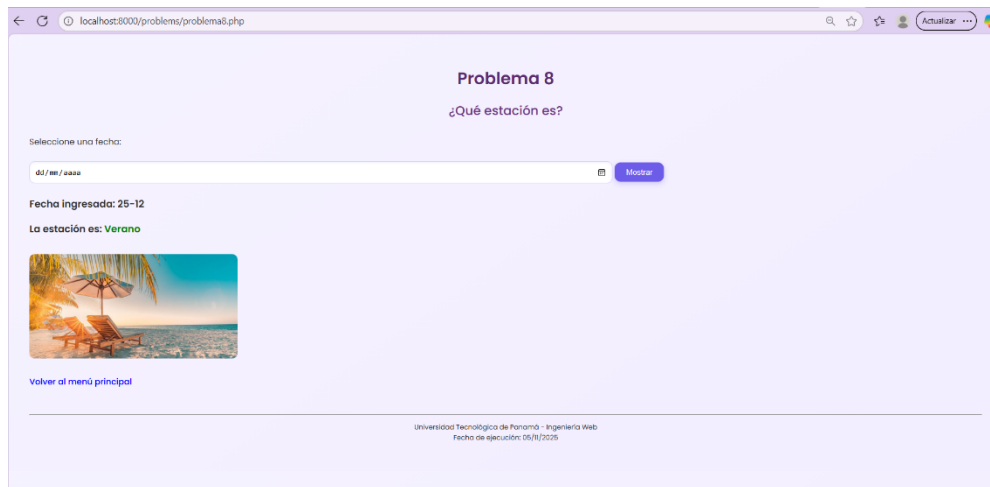


Figura 10. Problema 9 – Potencias del número ingresado

Aplicación que genera y muestra las 15 primeras potencias del número proporcionado (entre 1 y 9). La segunda parte del resultado se representa mediante una gráfica de línea que evidencia el crecimiento exponencial de los valores obtenidos.

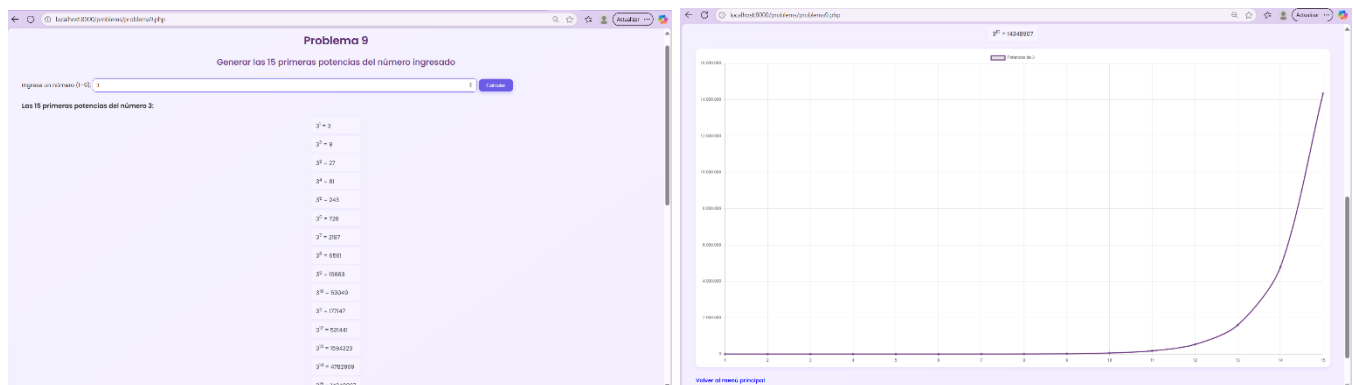


Figura 11. Problema 10 – Ventas de frutas por vendedor

Formulario que registra las cantidades vendidas de diferentes productos por cada vendedor. El sistema calcula el total de ventas por producto y por vendedor, mostrando los resultados en una tabla dinámica y ordenada.

←↻🔍🌟👤Actualizar ...

Problema 10

Ventas de Frutas por Vendedor

Ingrese la cantidad vendida de cada fruta por cada vendedor:

Fruta	Precio (USD)	Vendedor 1	Vendedor 2	Vendedor 3	Vendedor 4
Manzana	\$1.25	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Banano	\$0.75	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Mango	\$1.50	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Piña	\$1.80	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Sandia	\$2.00	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

Calcular totales

Tabla de Ventas Mensuales (USD)

Fruta	Precio Unitario	Vendedor 1	Vendedor 2	Vendedor 3	Vendedor 4	Total Fruta
Manzana	\$1.25	\$2.50	\$3.75	\$1.25	\$5.00	\$12.50
Banano	\$0.75	\$0.00	\$3.00	\$0.00	\$3.75	\$6.75
Mango	\$1.50	\$7.50	\$0.00	\$0.00	\$0.00	\$7.50
Piña	\$1.80	\$1.80	\$0.00	\$1.80	\$0.00	\$3.60
Sandia	\$2.00	\$14.00	\$20.00	\$14.00	\$18.00	\$66.00
Total por Vendedor		\$25.80	\$26.75	\$17.05	\$26.75	-

Volver al menú principal

Universidad Tecnológica de Panamá - Ingeniería Web
Fecha de ejecución: 05/11/2025

En cada programa se incluye en su parte inferior el pie de página con la fecha de ejecución y el nombre de la Universidad Tecnológica de Panamá, que confirma la correcta integración del componente externo solicitado en el enunciado del proyecto.