

# Fundamentos de Programación

## Evaluación del curso.

- Individual
- Proyectos y prácticas 100%
  - Se suben a Github (Repositorios privados y con invitación)
  - El trabajo se evalúa en base a cumplimiento de requerimientos
  - Posteriormente hay una evaluación personal y puede modificar la nota del trabajo

Contacto:

Tel. 614 153 52 97

[fer\\_manzanares@hotmail.com](mailto:fer_manzanares@hotmail.com)

## Práctica 1.

Desarrollar un programa en lenguaje Java que calcule para su usuario las operaciones aritméticas básicas: suma, resta, multiplicación y división a partir del ingreso de los valores de los dos operandos.

## Práctica 2.

Desarrollar un programa en lenguaje Java que calcule el equivalente en pies de una medición hecha en metros.

## Instrucciones repositorio github

### ...or create a new repository on the command line

```
echo "# serv_web2025" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/fer-manzanares/serv_web2025.git
git push -u origin main
```

## Primer proyecto del curso.

Se desea construir un **Convertor Universal de Unidades** que permita convertir entre distintas magnitudes (longitud, masa, temperatura, volumen, velocidad, entre otras).

El usuario podrá:

- Seleccionar el tipo de magnitud (p. ej. "Longitud").
  - Elegir la unidad de origen y destino (p. ej. de "Metros" a "Pulgadas").
  - Ingresar un valor y obtener el resultado de la conversión.
- 

### Requerimientos funcionales

#### 1. Interfaz gráfica principal con:

- Un *ComboBox* o un *TabPanel* para seleccionar la magnitud (Longitud, Masa, Temperatura, Volumen).
- Dos *ComboBox* para elegir la unidad de origen y destino.
- Un campo de texto para ingresar el valor.
- Un botón "Convertir".
- Una etiqueta o campo no editable donde mostrar el resultado.

#### 2. Conversión correcta entre unidades predefinidas, por ejemplo:

**Longitud:** metros, kilómetros, millas, pulgadas.

**Masa:** kilogramos, libras, gramos.

**Temperatura:** Celsius, Fahrenheit, Kelvin.

**Volumen:** litros, mililitros, galones.

#### 3. Validaciones:

- Solo se aceptan valores numéricos.
  - Manejo de errores (por ejemplo, si no se introduce valor o si hay una unidad no compatible).
-

## Criterios de evaluación

Criterio	Descripción	Ponderación
Diseño modular	Separación clara entre lógica y GUI	10%
Funcionalidad de conversión	Conversión correcta entre unidades, número de unidades y de conversiones	60%
Interfaz gráfica	Diseño limpio, intuitivo y funcional	10%
Validaciones y manejo de errores	Evita fallos ante entradas incorrectas	20%

---

## Extensiones opcionales (Puntos extra)

- Añadir un historial de conversiones.
- Incorporar selector de idioma (Español/Inglés).
- Mostrar fórmulas usadas en cada conversión.
- Implementar tema claro/oscuro.

## Segundo proyecto del curso.

Se desea construir una **Calculadora de Ventas y Utilidades** que determine los ingresos y utilidades semanales de un negocio. A continuación una descripción de lo que podrá hacer el usuario.

En una tabla:

- Ingresar la lista de productos que vende
- Ingresar por día de la semana en curso la cantidad de unidades vendidas de cada producto, incluyendo unidades decimales para productos que se venden a granel

En una segunda tabla que se corresponda a nivel de fila con la primera:

- El precio de compra (costo) de cada producto
- El precio de venta de cada producto

En una tercer tabla que se corresponda a nivel de fila con la primera:

- Los importes totales de ventas de la semana de cada producto
- Los importes totales de utilidades de la semana de cada producto

En tiempo de ejecución del programa el usuario podrá definir cuántos productos comercializa para crear el número adecuado de filas.

Cada tabla debe incluir una columna para el nombre del producto. El nombre del producto debe poderse ingresar en la primer tabla y en automático debe llevarse a las filas correspondientes de las otras dos tablas.

La primer tabla debe tener otras siete columnas para cada uno de los días de la semana.

La segunda tabla tendrá, como se menciona previamente, dos columnas más: una para el costo o precio de compra y otra para el precio de venta.

La tercer tabla es para cálculos solamente, por lo que no es editable. Tendrá otras dos columnas: una para el total de ingresos y otra para el total de utilidades por semana.

Las tres tablas deben tener títulos para cada columna. (La primera por ejemplo: Producto Lunes Martes Miércoles Jueves Viernes Sábado Domingo).

## Requerimientos funcionales

### 1. Interfaz gráfica principal con:

- JTables para ingresar datos y presentar resultados.
- un JFormattedTextfield y un JButton para definir el número de productos a manejar.
- Un JButton para iniciar el cálculo y presentar resultados.
- Un JButton para guardar la información.
- Un JButton para cerrar la aplicación.

### 2. Cálculo correcto ingresos y utilidades

### 3. Validaciones:

- Solo se aceptan valores numéricos positivos (se incluye el cero).
- Manejo de errores.

## Criterios de evaluación

Criterio	Descripción	Ponderación
Diseño modular	Separación clara entre lógica y GUI	10%
Funcionalidad de conversión	Conversión correcta entre unidades, número de unidades y de conversiones	50%
Interfaz gráfica	Diseño limpio, intuitivo y funcional	30%
Validaciones y manejo de errores	Evita fallos ante entradas incorrectas	10%

## Extensiones opcionales (Puntos extra)

- Limpiar los valores de las tablas sin eliminar los nombres de los productos para iniciar otra semana.
- Implementar tema claro/oscuro.