

## Proyecto mínimo para los tipos de pruebas (UT - 2.4)

Por parejas o tríos, usando la teoría de la UT 2 (punto 4) tendréis que realizar un app mínima con su testing correspondiente.

Únicamente nos centraremos en el punto 4 de la UT 2 por el momento.

**Podéis elegir el lenguaje que prefiráis.**

Creareis un **proyecto mínimo funcional** que cuente con:

1. Pruebas **unitarias** de funciones básicas.
2. Pruebas **de integración** que combinen dos módulos o más.
3. Pruebas **de sistema (E2E)** que simulen un uso real.
4. Prueba **de humo** para verificar que la app arranca.

### Tareas

1. **Preparar el entorno local**
  - Crear desde cero un proyecto simple (calculadora o conversor).
  - Configurar **Makefile** con el comando **make test**.
2. **Desarrollar las funciones base**
  - Por ejemplo: **sumar()**, **restar()**, **dividir()**, **promedio(lista)**.
3. **Implementar las pruebas unitarias**
  - Debes validar entradas normales y casos límite.
  - Si hay errores esperados, usar **pytest.raises()** o **toThrow()**.
4. **Crear las pruebas de integración**
  - Combinar lectura de archivo, función de cálculo y salida formateada.
  - Verificar que los resultados se calculan correctamente.

## 5. Crear las pruebas E2E / sistema

- Implementar un pequeño CLI (interfaz por consola).
- Ejecutar el programa desde el test con `subprocess` (Python) o `child_process` (JS).

## 6. Añadir la prueba de humo

- Importar el módulo principal o iniciar el programa y comprobar que no lanza errores.

## 7. Ejecutar toda la suite y generar informe de cobertura

- Objetivo:  $\geq 60\%$  de cobertura.
- Capturar la salida de terminal como evidencia.

## 8. Documentar el trabajo

- Añadir un `README.md` con breve descripción, estructura de carpetas y tabla de pruebas

## Entregables

Los proyectos los revisaremos en clase. Los proyectos subidos a github y añadirme como colaborador

PVallejoEduca

pvallejgon@educacion.navarra.es