

Escuela Politecnica Nacional

Facultad de Ingeniería de Sistemas

Construcción y Evolución de Software

Versión:	1.0
Grupo:	5
Fecha:	Noviembre 2025

Casos de Prueba General

Este documento propone un conjunto de pruebas (unitarias, de integración y manuales) para verificar el correcto funcionamiento del proyecto.

1) Pruebas unitarias (función `texto_a_braille`)

Recomendado usar `pytest`. AGREGAR IMAGENES

- Caso: letras minúsculas
 - Entrada: `"abcxyz"`
 - Esperado: cada caracter mapeado correctamente según `BRaille_MAP`.
- Caso: números
 - Entrada: `"0123456789"`
 - Esperado: mapeo correcto (ver `BRaille_MAP`).
- Caso: puntuación y espacio
 - Entrada: `"Hola, mundo."`
 - Esperado: comas y puntos convertidos y espacios preservados.
- Caso: caracteres no soportados
 - Entrada: `"@#€"`
 - Esperado: los caracteres no mapeados se devuelven tal cual.
- Caso: cadena vacía
 - Entrada: `""`
 - Esperado: `""` (cadena vacía) o manejo definido.

Consideraciones

Ejemplo de test (esqueleto): SI ES OPCIONAL TDD

```
from app import texto_a_braille

def test_letras():
    assert texto_a_braille('abc') == '⠠⠠⠠'

def test_vacio():
    assert texto_a_braille('') == ''
```

3) Pruebas End-to-End (UI)

Manual o automatizado (Selenium / Playwright):

- Abrir `http://localhost:5000`.
- Introducir texto en el `textarea`.
- La traducción a texto braile es instantanea a la par que se introduce texto

4) Pruebas de accesibilidad

- Probar con lector de pantalla (NVDA, VoiceOver) y comprobar que las secciones son legibles.
- Comprobar contraste de colores y tamaño de fuente adaptables.

5) Pruebas de rendimiento

- Medir latencia para textos de diferentes tamaños (por ejemplo: 1KB, 10KB, 100KB).
- Determinar throughput concurrente con herramientas como `ab` o `wrk` si se despliega detrás de un servidor WSGI.

6) Pruebas de seguridad

- Limitar tamaño de payload para prevenir DoS.
- Validar que la API no ejecuta código o realiza operaciones peligrosas con la entrada.

Checklist de aceptación

- Conversión correcta para letras básicas.
- API devuelve 400 para entradas vacías.
- UI muestra y copia correctamente el resultado.
- Aplicación arranca en entorno virtual con `pip install -r requirements.txt`.
- Pruebas unitarias y de integración pasan en CI.