

Asignatura	Datos del alumno	Fecha
<b>Entornos de Integración y Entrega Continua</b>	Apellidos:	
	Nombre:	

## Actividad individual: pruebas en Python

### Objetivos

En esta actividad practicaréis con la escritura de pruebas en Python. Aprenderéis a distinguir entre prueba unitaria y prueba de sistema.

### Pautas de elaboración

El código del repositorio `unir-test`, utilizado en el caso práctico, contiene una batería de pruebas insuficiente y, además, no ha implementado todas las funcionalidades. Deberán completar las funcionalidades que faltan y añadir más pruebas para cubrir el mayor número de funciones posibles.

Deberán modificar los ficheros `api.py` y `calc.py` para que incluyan las funciones de suma, resta, multiplicación, división, potenciación, raíz cuadrada y logaritmo en base 10. Algunas de las funciones necesarias ya están implementadas.

Las funciones de la clase `Calculator` deben comprobar que los parámetros tengan valores aceptables. Por ejemplo, la división, ya implementada, eleva una excepción si el divisor es 0. Podéis usar `TypeError` para todas las excepciones de esta clase.

Asignatura	Datos del alumno	Fecha
<b>Entornos de Integración y Entrega Continua</b>	Apellidos:	
	Nombre:	

Añadid pruebas unitarias en `unit/calc_test.py` y `rest/api_test.py`. Si has definido funciones auxiliares, añade las pruebas en `unit/util_test.py`. Las pruebas deben cubrir los casos de éxito y los casos de fallo. Por ejemplo, una llamada `GET /calc/divide/1/0` debe comprobar que la API devuelve un código HTTP 400 Bad Request. Las pruebas de la clase `Calculator` deben cubrir métodos estáticos, si se define algunos.

Aprovechad el módulo `math` para las funciones matemáticas avanzadas. Si lo necesitáis, refactoriza la clase `Calculator` para reutilizar código.

### Extensión y formato

Los cuatro ficheros Python y los informes de resultados en XML de las pruebas unitarias y de API. Si has definido funciones auxiliares, añade también los ficheros correspondientes.

### Rúbrica

Actividad individual: pruebas en Python	Descripción	Puntuación máxima (puntos)	Peso %
Criterio 1	R) La clase <code>Calculator</code> contiene todas las funciones y la API contiene todos los <i>endpoints</i> .	3	30%
Criterio 2	Las pruebas unitarias y las de API cubren todas las funciones.	3	30%

Asignatura	Datos del alumno	Fecha
<b>Entornos de Integración y Entrega Continua</b>	Apellidos:	
	Nombre:	

Criterio 3	Las pruebas unitarias y de API cubren casos de éxito y casos de error.	4	40%
		<b>10</b>	<b>100 %</b>