

Calculadora

Pruebas Unitarias

versión [3]

Fecha de realización de las pruebas: 28 de septiembre, 2025.

Introducción

En el desarrollo de la aplicación Calculadora V3, las pruebas unitarias son un paso fundamental para asegurar la calidad y fiabilidad del software. Estas pruebas se centran en las funciones más pequeñas del código, como las operaciones aritméticas básicas, el cálculo de porcentaje, y las funciones de memoria e historial. Al verificar cada componente de forma aislada, podemos detectar errores en las primeras etapas del desarrollo, garantizando que cada pieza del software funcione correctamente antes de ser integrada en el sistema completo. Aunque las pruebas unitarias suelen ser ejecutadas por los desarrolladores, la colaboración del equipo de pruebas es clave para definir los casos de prueba y asegurar que los resultados de cada unidad cumplan con los requerimientos funcionales del proyecto.

Especificación de la Unidad

Para la Calculadora V3, cada funcionalidad principal se considera una unidad de software independiente. A continuación, se detalla la especificación de cada una:

- **Unidad Suma:** Recibe dos números y devuelve su suma.
 - **Entrada:** a, b
 - **Proceso:** $a+b$
 - **Salida esperada:** El resultado de la suma.
- **Unidad Resta:** Recibe dos números y devuelve su diferencia.
 - **Entrada:** a, b
 - **Proceso:** $a-b$
 - **Salida esperada:** El resultado de la resta.
- **Unidad Multiplicación:** Recibe dos números y devuelve su producto.

- **Entrada:** a, b
- **Proceso:** $a \times b$
- **Salida esperada:** El resultado de la multiplicación.
- **Unidad División:** Recibe dos números y devuelve su cociente.
 - **Entrada:** a, b
 - **Proceso:** $a \div b$
 - **Salida esperada:** El resultado de la división, gestionando el caso donde $b \neq 0$.
- **Unidad Porcentaje:** Recibe un número y calcula su porcentaje basado en otro valor.
 - **Entrada:** número, porcentaje
 - **Proceso:** $(\text{número} \times \text{porcentaje}) \div 100$
 - **Salida esperada:** El valor del porcentaje calculado.
- **Unidad de Memoria (MC, MR, M+, M-):** Gestionan el almacenamiento, recuperación y modificación de un valor numérico en memoria.
 - **MC (Memory Clear):**
 - **Entrada:** Acción del usuario (botón MC).
 - **Proceso:** Borra cualquier valor almacenado en la memoria (lo establece en 0).
 - **Salida esperada:** La memoria queda vacía o con valor 0.
 - **MR (Memory Recall):**
 - **Entrada:** Acción del usuario (botón MR).
 - **Proceso:** Recupera el valor almacenado en la memoria.
 - **Salida esperada:** El número guardado en la memoria se muestra en la pantalla principal.
 - **M+ (Memory Add):**
 - **Entrada:** El número actual en pantalla.
 - **Proceso:** Suma el número de la pantalla al valor ya existente en la memoria.
 - **Salida esperada:** El valor en memoria se actualiza con el resultado de la suma.

- **M- (Memory Subtract):**
 - **Entrada:** El número actual en pantalla.
 - **Proceso:** Resta el número de la pantalla al valor ya existente en la memoria.
 - **Salida esperada:** El valor en memoria se actualiza con el resultado de la resta.
- **Unidad de Historial:** Registra y muestra las operaciones realizadas previamente.
 - **Entrada:** Una operación completada.
 - **Proceso:** Almacenamiento de la operación y su resultado en una lista.
 - **Salida esperada:** El historial de operaciones se actualiza y muestra correctamente.

Objetivo de la Unidad

El objetivo principal de probar cada unidad es validar que cada función se comporte de acuerdo con los requerimientos definidos, resolviendo una tarea específica de forma aislada y correcta. Con esto buscamos:

- **Confirmar** que las operaciones aritméticas (suma, resta, multiplicación y división) devuelvan resultados precisos con números enteros, decimales y negativos.
- **Asegurar** que la función de división gestione adecuadamente el caso de división por cero, mostrando un error controlado y sin interrumpir la aplicación.
- **Verificar** que la lógica de entrada de datos solo permita **un punto decimal por cifra** para evitar números malformados (ej. 5.3.2).
- **Validar** que el sistema acepte y opere correctamente con números de hasta un **máximo de dos decimales**.
- **Verificar** que el cálculo de porcentaje sea correcto y se integre adecuadamente con las demás operaciones.
- **Validar** que las funciones de memoria (MC, MR, M+, M-) y el sistema de historial operen de manera fiable e independiente.
- **Reducir** la probabilidad de encontrar defectos en etapas posteriores, como las pruebas de integración y de sistema, facilitando así el mantenimiento del código.

Pruebas

Unidad	Entradas	Proceso	Salida esperada	Tipo de caso	Estado
Suma	(5, 3)	$5 + 3$	8	Normal	Ok
Suma	(39.02, 57.43)	$39.02 + 57.43$	96.45	Normal	Ok
Suma	(-10, 2)	$-10 + 2$	8	Números negativo	Ok
Resta	(22, 3)	$22 - 3$	19.00	Normal	Ok
Resta	(289.89, 524.46)	$289.89 - 524.46$	-234.57	Número negativo con punto decimal	Ok
División	(120, 2)	$120 \div 2$	60	Normal	Ok
División	(720, 184)	$720 \div 184$	3.91	Con punto decimal	Ok
División	(52.46, 12.35)	$52.46 \div 12.35$	4.25	Con punto decimal	Ok
División	(10, 0)	$10 \div 0$	Math ERROR	Caso de error	Ok
Multiplicación	(300, 3)	300×3	900	Normal	Ok
Multiplicación	(3.29, 172.46)	3.29×172.46	567.39	Con punto decimal	Ok
Multiplicación	(-3, 2)	-3×2	-6	Número negativo	Ok
Porcentaje	(5%)	$5 \div 100$	0.05	Normal	Falla, requiere multiplicarse por 1 culminar correctamente.
Porcentaje	(-10%)	$-10 \div 100$	-0.10	Número negativo	Falla, requiere multiplicarse por 1 culminar correctamente.
Porcentaje	(12.22%)	$12.22 \div 100$	0.12	Con punto decimal	Falla, requiere multiplicarse por 1 culminar correctamente.
Memoria	(25 M+, 10 M+, 30 M+, MR)	En memoria: $25 + 10 + 30$	65	Normal	OK

Memoria	(12.05 M+, 124.72 M+, MR)	En memoria: 12.05 + 124.72	136.77	Con punto decimal	Ok
Memoria	(25 M-, 10 M-, 25 M-, MR)	En memoria: - 25 - 10 - 25	-60	Números negativos	Ok
Memoria	(34.27 M-, 362.59 M-, MR)	En memoria: - 34.27 – 362.59	-396.86	Número negativo con punto decimal	Ok
Memoria	(25 M+, MC, 12M+, MR)	En memoria: 12	12	Normal	Ok
Historial	(6 x 5)	6 x 5	6 x 5 = 30	Normal	Ok
Historial	(12.42 + 65.13)	12.42 + 65.13	12.42 + 65.13 = 77.55	Con punto decimal	Ok
Historial	(3 x 8, 24 + 2, 26 - 7, 19 ÷ 4, 12%)	3 x 8 24 + 2 26 – 7 19 ÷ 4 12 ÷ 100	3 x 8 = 24 24 + 2 = 26 26 – 7 = 19 19 ÷ 4 = 4.75 12% = 0.12	Operaciones continuas	Falla, el porcentaje no aparece como se esperaba.

Registro de Incidencias

ID Incidencia	Descripción	Severidad	Estado	Responsable
I-001	Al realiza una operación donde primero se coloque un número, luego el operador “x”, luego el operador “÷”, después otro número y por último el operador “=”, la calculadora lanza el resultado de cero	Alta	Pendiente	Desarrollador
I-002	Al realiza una operación donde primero se coloque un número, luego el operador “x”, luego el operador “+”, después otro	Alta	Pendiente	Desarrollador

	número y por último el operador "=", la calculadora lanza como resultado el último número ingresado. Pasa lo mismo con el operador "-" después de colocar el operador "x".			
I-003	Al terminar de realizar una operación y presionar el operador "=", de realiza una nueva operación entre el número del resultado (como número 1 y 2) con el operador utilizado en la operación anterior.	Alta	Pendiente	Desarrollador
I-004	Al haber realizado una operación, luego borrarla, colocar cualquier número y el operador "=", se realiza una operación entre el número realizado (como segundo número) y "0" (como primer número) con el último operador utilizado.	Alta	Pendiente	Desarrollador
I-005	La calculadora permite colocar números después de haber colocado el operador "%".	Alta	Pendiente	Desarrollador
I-006	Al realizar el cambio de operador, se guarda en el historial una operación con el operador cambiado y colocando el segundo número como "0".	Medio	Pendiente	Desarrollador
I-007	Cuando el primer número es negativo, en pantalla no se	Medio	Pendiente	Desarrollador

	visualiza el signo “-” en frente del número colocado.			
I-008	La calculadora puede mostrar un máximo de 17 números, apareciendo el último número cortado	Baja	Pendiente	Desarrollador
I-009	El historial muestra solo las ultimas 5 operaciones realizadas, al realizar la sexta, la primera operación guardada se reescribe	Baja	Pendiente	Desarrollador