



Ficha de ayuda: Diseño de pruebas

El diseño de pruebas consiste en crear **casos de prueba**. Cuando realizaste análisis de requisitos, te preguntas "¿Qué debo probar?" Cuando creas casos de prueba, te preguntas "**¿Cómo hago la prueba?**". Es como escribir los pasos para cocinar un emocionante plato nuevo.

Reglas para el diseño de pruebas

Hay cinco reglas importantes a la hora de diseñar pruebas.

1. **Podrías necesitar varios casos de prueba para verificar un requisito.** Por ejemplo, tienes un requisito que dice: "El mapa de Urban Routes es escalable". Puedes probar esta funcionalidad de diferentes maneras: utilizando los botones "+" y "-" de la interfaz, o con un mouse, un panel táctil o mediante movimientos en dispositivos móviles.
2. **Un caso de prueba equivale a una prueba.** Si pruebas la escalabilidad del mapa de Urban Routes, sería conveniente crear dos casos de prueba: uno para el botón "+" y otro para el botón "-". Si se combinan ambas pruebas en un solo caso y uno de los botones tiene un bug, no sabrás cuál es el responsable.
3. **Las pruebas no deben duplicar la funcionalidad.** Supongamos que has probado el botón "+" en el caso de prueba "Escalabilidad del mapa Urban Routes". Significa que no necesitas un caso de prueba para "Visualización del botón '+' en la interfaz". Obviamente, el botón aparece.
4. **Diseña tus pruebas dentro de los requisitos.** Si en los requisitos no se describe cómo funcionará un smartphone a -120 °C, no escribas un caso de prueba en el que necesites sumergir el smartphone en nitrógeno líquido.

5. **Las pruebas deben cubrir todos los requisitos.** Debes asegurarte de que todos los casos de prueba comprueben todas las funciones.

Cuándo terminar el diseño de la prueba

Utiliza las reglas anteriores para saber cuándo has terminado el diseño de la prueba. Si los casos de prueba cubren todos los requisitos y no se duplican entre sí, has terminado. Esa es la definición de "terminado".

Pruebas positivas y negativas

El diseño de las pruebas implica la creación de casos de prueba positivos y negativos.

Un **caso de prueba positivo** verifica que una aplicación funcione de manera correcta de acuerdo con un requisito particular y cuando se usen datos válidos.

Los **casos de prueba negativos** verifican que la aplicación responda con éxito cuando se usa de manera incorrecta o cuando se utiliza un dato no válido. La aplicación debe lograr manejar con éxito un error para responder frente un comportamiento inesperado del usuario.

Orden de las pruebas

En primer lugar, hay que hacer pruebas positivas. Son más importantes a medida que compruebas la aplicación de los requisitos.

Equivalencia

Una de las técnicas fundamentales del diseño de pruebas es la partición de clases de equivalencia.

¿Qué es la equivalencia?

La **equivalencia** es la igualdad de los objetos. En las pruebas, los valores equivalentes son aquellos que una aplicación procesa de la misma manera.

Por ejemplo, los requisitos para los campos de dirección en Urban Routes afirman: "La longitud aceptada es de 1 a 50 caracteres".

Esto quiere decir que la aplicación procesa los valores de 1 a 50 caracteres de longitud de la misma manera. Los valores con longitudes mayores que 50

caracteres se procesan diferente, por ejemplo, mediante un mensaje de error. El equipo de testers combina estos valores en **clases de equivalencia**.

¿Para qué sirven las clases de equivalencia?

En lugar de realizar muchas pruebas para cada grupo, puedes realizarle la prueba a un valor de cada grupo. La aplicación reaccionará a cualquiera de ellos de la misma manera. Puedes escoger los siguientes valores:

- 0 caracteres (dejar el campo vacío genera un resultado no válido).
- 20 caracteres (20 es más de 1 y menos de 50, por lo tanto, debe ser válido).
- 52 caracteres (52 es más de 50, por lo tanto, debe ser no válido).

Además de ahorrar tiempo a la hora de redactar y realizar los casos de prueba, esta técnica permite cubrir todos los posibles escenarios, lo que hace que sean pruebas exhaustivas.

Cómo identificar las clases de equivalencia

Utiliza el siguiente algoritmo para identificar las clases de equivalencia:

1. Descompón el requisito en bloques atómicos.
2. Para cada bloque, identifica el tipo de clase: un rango de números o un conjunto de valores.
3. Identifica los valores correctos. Puedes encontrarlos en los requisitos.
4. Identifica los valores incorrectos. Incluyen todo lo que no se identificó en la segunda etapa.
5. Descubre lo que ocurre si introduces valores incorrectos.

Un rango y un conjunto de valores

Una clase de equivalencia puede ser un rango o un conjunto de valores.

- Un **rango** es un intervalo de números con límites, por ejemplo, 1–50.
- Un **conjunto de valores** es un grupo de valores que no se puede dividir en intervalos. Por ejemplo, cuando seleccionas un idioma de una lista

desplegable en un sitio web, un conjunto comprenderá cada idioma que selecciones. Otro ejemplo es el siguiente: "Una contraseña debe contener caracteres especiales". En este ejemplo, la clase "Contraseña con caracteres especiales" será un conjunto porque no puedes dividir sus valores de prueba en intervalos.

La aplicación debe reaccionar ante el mismo rango o el mismo conjunto de valores de la misma manera.

Cómo elegir los valores de prueba

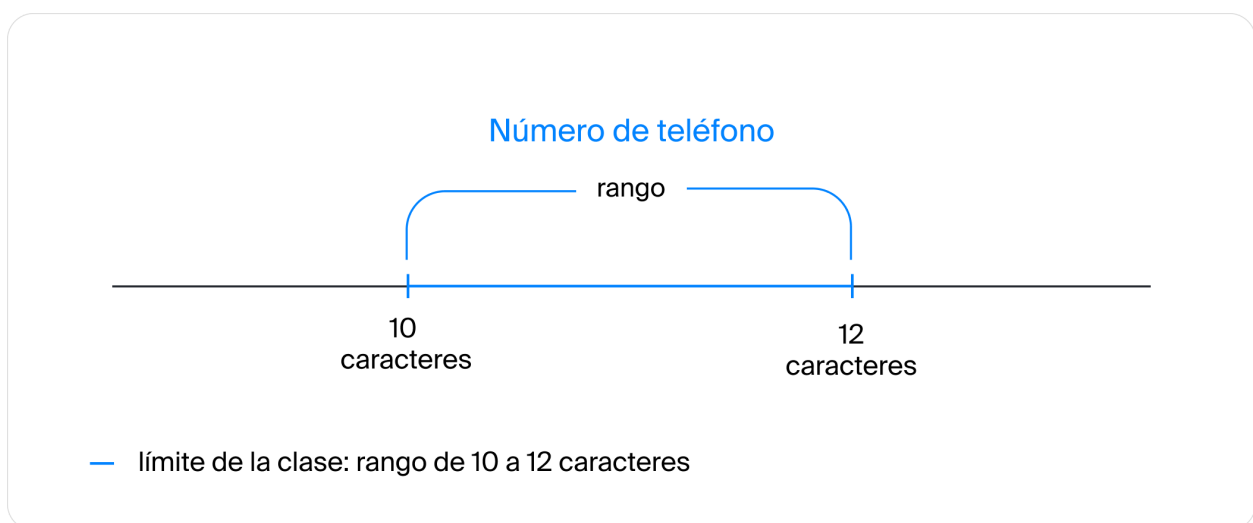
Existe una hipótesis sobre hacer pruebas: si una aplicación procesa correctamente un valor de una clase, hará lo mismo con el resto de los valores.

Para encontrar un valor de prueba, elige un valor dentro de una clase.

Valores límite

Un **valor límite** es un valor equivalente al valor mínimo o máximo permitido.

Por ejemplo, los requisitos indican que un número de teléfono puede contener de 10 a 12 caracteres, es decir, 10 y 12 son los límites de la clase.



Ten en cuenta que solo un rango puede tener valores límite. Los conjuntos no los tienen.

Si una clase es un conjunto, los requisitos suelen describir todos los valores de este conjunto. Por ejemplo, un usuario puede seleccionar alguna de las siguientes tarifas de taxi: "Laboral", "Sueño", "Relajante", "Hablador", "Comfort" o "Glamuroso".

Cómo encontrar los límites

Se permiten varios tipos de datos en los rangos:

- números enteros (1, 2, 3...)
- números decimales (0.1; 0.2; 0.3...)
- intervalos de tiempo (horas, días, meses)

Para identificar los valores límite, debes estudiar los requisitos. Los requisitos pueden definir los límites de diferentes maneras:

- **Intervalo numérico claro:** cuando sabemos los valores mínimos y máximos exactos, como "De 10 a 12 caracteres".
- **Rango claro:** cuando sabemos el rango, pero no sabemos los valores exactos, como "Números positivos".
- **Intervalo numérico poco claro:** cuando tenemos valores aproximados que debemos esclarecer, por ejemplo, "Mayoría de edad".
- **Rango poco claro:** cuando tenemos un rango aproximado con límites sin definir, por ejemplo, "Una persona de cualquier edad puede obtener un pasaporte".

Diseño de pruebas

Por ejemplo, solo los adultos pueden recibir un préstamo. Según las normas del banco, su edad debe estar entre los 18 y los 65 años.

Para conocer las oportunidades de préstamo, tienes que introducir tu edad en un campo específico de la página web del banco.

Cómo comprobar el funcionamiento de una aplicación con valores límite:

1. Identifica el rango de valores, es decir, de 18 a 65 años, ambos incluidos. En este rango, una persona es mayor de edad, pero sin llegar a ser pensionista.
2. Comprueba que la aplicación funcione con los valores límite requeridos, es decir, 18 y 65.
3. Comprueba que la aplicación funcione en función de una lógica diferente cuando está fuera de los límites del rango, es decir, con valores de 17 y 66.
4. Comprueba que la aplicación funcione como se requiere cuando está dentro de los límites del rango, es decir, con valores de 19 y 64.

Así, las pruebas del campo incluyen seis comprobaciones: 18 y 65; 17 y 66; 19 y 64.

Algoritmo general para el diseño de pruebas basadas en valores límite

Supongamos que tienes que comprobar que el minuterio de un reloj funciona como se requiere en un intervalo de 15 a 45 minutos. Tipo de datos: intervalo de tiempo. Intervalo: 1 minuto.

1. Identifica el tipo de datos y el intervalo: un intervalo de tiempo y 1 minuto.
2. Comprueba que la aplicación funciona bien con los valores límite del rango: 15 y 45.
3. Muévete un intervalo fuera del rango y comprueba los siguientes valores: 14 y 46.
4. Muévete un intervalo dentro del rango y comprueba los siguientes valores: 16 y 44.

Cómo utilizar clases de equivalencia y valores límite juntos

¿Cuál sería la mejor forma de comprobar la recarga del saldo (del ejemplo de la lección) mediante las técnicas de las clases de equivalencia y los valores límite? Primero, tienes que probar los valores dentro de la clase y, luego debes utilizar el algoritmo para probar los valores límite:

1. **Prueba los valores dentro de las clases:** 00.50, 2.50 y 55.00. Si las pruebas son satisfactorias, ve al siguiente paso. Si no, no tiene sentido seguir probando la aplicación. No funciona.
2. **Prueba los valores límite:** 0.00 y 00.99; 1.00 y 49.99; 50.00. Si las pruebas con estos valores son exitosas, la aplicación funciona bien. La última clase carece de límite máximo, por lo que no es necesario probarla.
3. **Prueba los valores un intervalo fuera de los límites:** 1.00; 0.99 y 50.00; 49.99. Si las pruebas son satisfactorias, habrás confirmado que los límites son correctos. No se puede establecer un valor fuera de los límites de la primera clase: no es posible recargar el saldo con una suma negativa.
4. **Prueba los valores un intervalo dentro de los límites:** 0.01 y 0.98; 1.01 y 49.98; 50.01. Si las pruebas son exitosas, significa que la aplicación funciona como se requiere dentro de todo el rango; por lo tanto, el sistema reacciona a los valores de una clase de la misma manera.

Intenta recordar el orden de aplicación de las clases de equivalencia y las técnicas de valores límite:

1. En primer lugar, hay que identificar las clases de equivalencia.
2. Identifica los valores límite de las clases que tengan un formato de rango.
3. Forma los valores de prueba para todas las clases, tanto para los rangos como para los conjuntos de valores.

Optimización de las pruebas

El principio más importante de optimización es reducir el número de duplicados antes de las pruebas.

Situaciones complicadas

Hay situaciones en las que dos requisitos de la documentación se solapan.

Un banco permite transferir dinero de una tarjeta a otra, siguiendo dos condiciones:

1. No puedes transferir menos de 5 dólares ni más de 5,000 dólares de una vez.
2. El importe de la transferencia no puede superar el saldo de la cuenta.

Ambos requisitos deben cumplirse al mismo tiempo. Puedes probar el servicio teniendo en cuenta dos requisitos, siempre y cuando dividas el intervalo en clases realizando varias etapas.

Teniendo en cuenta la primera condición, puedes dividir el intervalo en tres clases con los siguientes rangos de valores:

N de la clase	Rango, \$	Comportamiento del sistema
Clase A	0 - 4.99	La transferencia es imposible
Clase B	5.00 - 5000.00	La transferencia es posible
Clase C	5000.01 - ∞	La transferencia es imposible

Las clases se han diseñado con la idea de que el saldo de la cuenta no está limitado.

Para cumplir el segundo requisito, hay que tomar un número del rango [5.0; 5000.00]. Por ejemplo, el saldo de la tarjeta es de solo 700 dólares. La clase B se dividirá en dos clases más con los rangos de valores B1 y B2.

N de la clase	Rango, \$	Comportamiento del sistema
Clase B1	5.00 - 700.00	La transferencia es posible
Clase B2	700.01 - 5000.00	La transferencia es imposible

Para comprobar cómo funciona el sistema con dos condiciones y optimizar el número de pruebas, hay que combinar ambos grupos de clases.

Presta atención a la columna "Saldo de la cuenta". Contiene las condiciones que hay que tener en cuenta al comprobar los rangos.

Rango, \$	Saldo de la cuenta	Comportamiento del sistema
0 - 4.99	sin límites	La transferencia es imposible
5.00 - 700.00	700	La transferencia es posible

Rango, \$	Saldo de la cuenta	Comportamiento del sistema
700.01 - 5000	700	La transferencia es imposible
5000.01 - ∞	sin límites	La transferencia es imposible

Has cumplido una de las principales condiciones del diseño de pruebas: cubrir los requisitos con un número mínimo de pruebas.