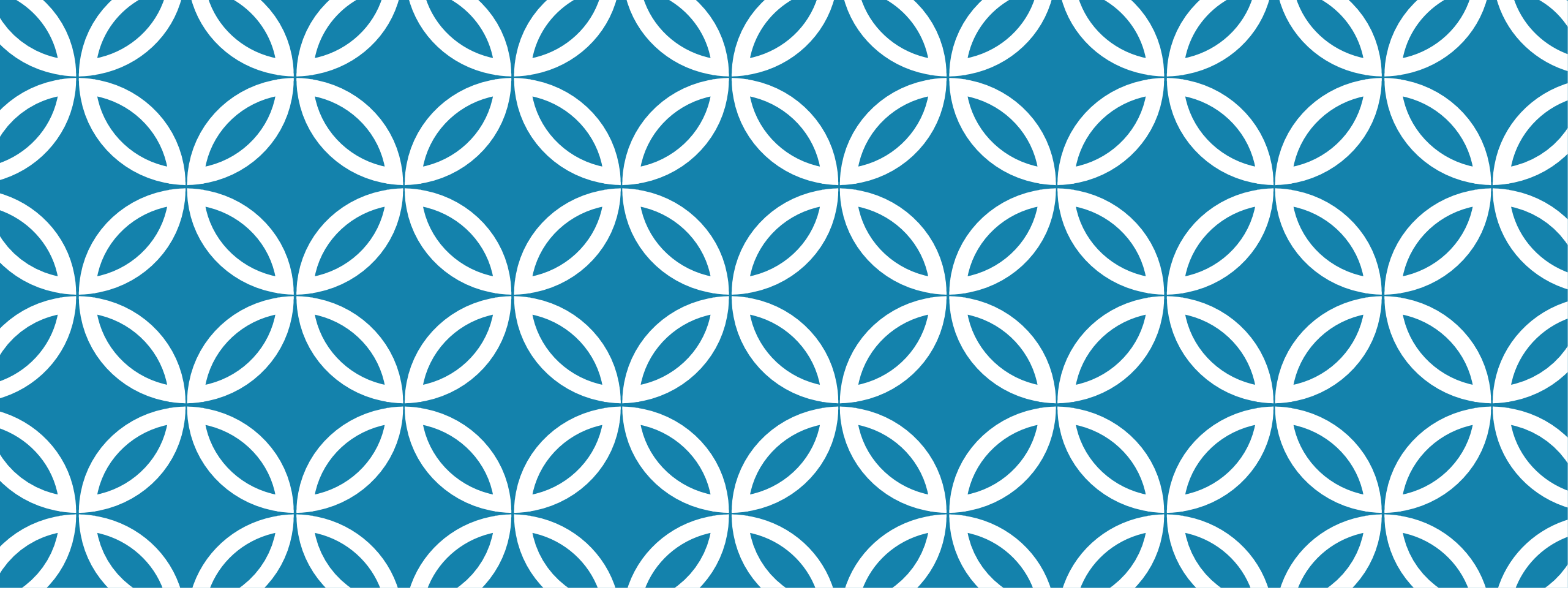


MÓDULO PROFESIONAL
ENTORNOS DE DESARROLLO

UD 4 – Diseño y realización de pruebas de caja negra

PRUEBAS DE CAJA NEGRA

- Técnicas de diseño de casos de prueba (caja negra):
  - Clases de equivalencia.
  - Análisis de valores límite.
  - Estudio de errores típicos.
  - Manejo de interfaz gráfica.
  - Datos aleatorios.

PRUEBAS DE CAJA NEGRA

Habrá elegir cuidadosamente los casos de prueba, de manera que sean tan pocos como sea posible para que la prueba se pueda ejecutar en un tiempo razonable y, al mismo tiempo, que cubran la variedad de entradas y salidas más amplia posible.

Para ello, se han **diseñado diferentes técnicas:**

- **Clases de equivalencia:** Se trata de determinar los diferentes tipos de entrada y salida, agruparlos y escoger casos de prueba para cada tipo o conjunto de datos de entrada y salida.
- **Análisis de los valores límite:** Estudian los valores iniciales y finales, ya que estadísticamente se ha demostrado que tienen más tendencia a detectar errores.
- **Estudio de errores típicos:** La experiencia dice que hay una serie de errores que se suelen repetir en muchos programas; por ello, se trataría de diseñar casos de prueba que provocaran las situaciones típicas de este tipo de errores.

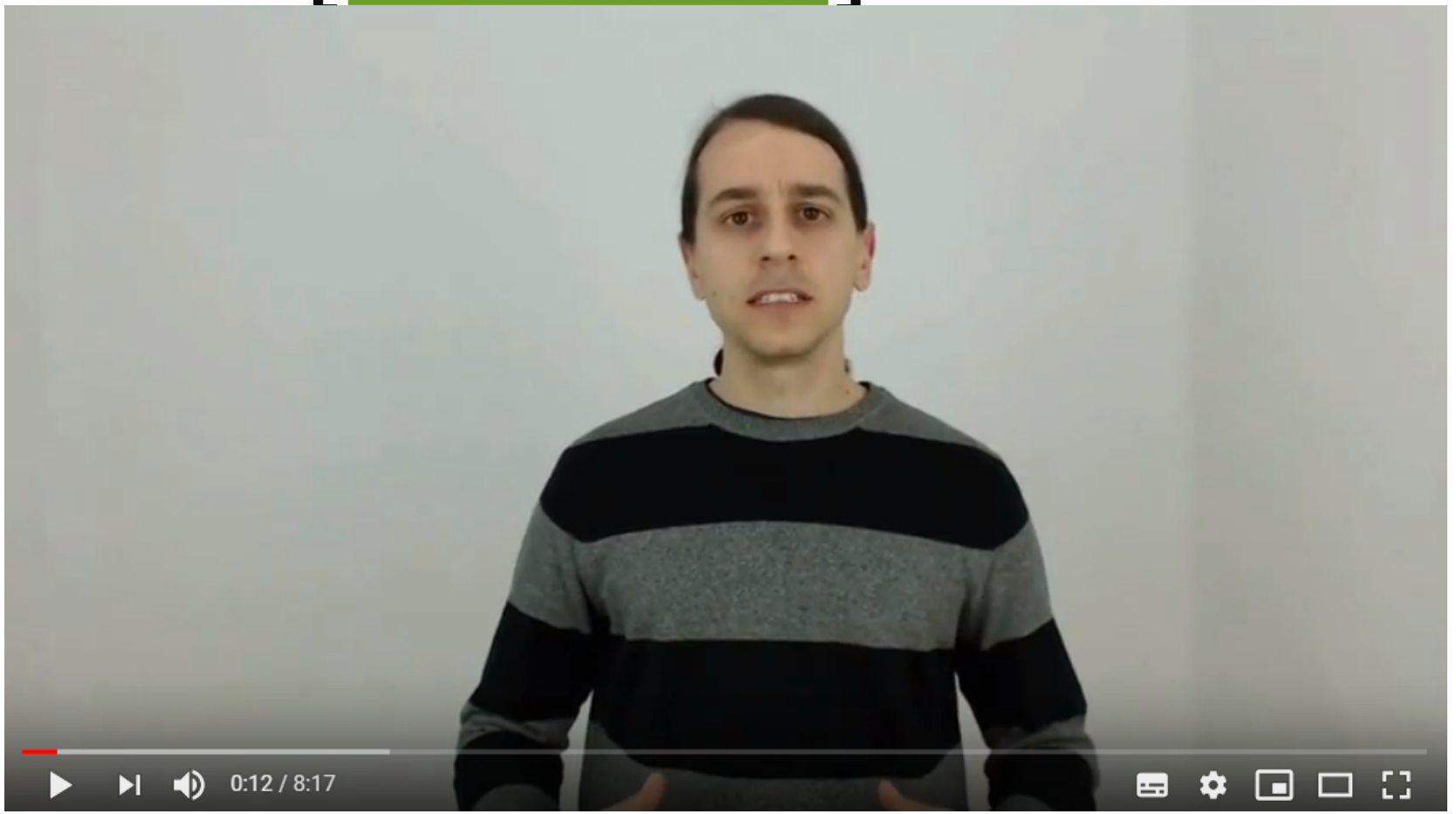
PRUEBAS DE CAJA NEGRA

- **Manejo de interfaz gráfica:** Para probar el funcionamiento de las interfaces gráficas, se deben diseñar casos de prueba que permitan descubrir errores en el manejo de ventanas, botones, iconos ...
- **Datos aleatorios:** Se trata de utilizar una herramienta que automatice las pruebas y que genere de forma aleatoria los casos de prueba. Esta técnica no optimiza la elección de los casos de prueba, pero si se hace durante bastante tiempo con muchos datos, podrá llegar a hacer una prueba bastante completa. Esta técnica se podría utilizar como complementaria a las anteriores o en casos en que no sea posible aplicar otra.

PRUEBAS DE CAJA NEGRA

Un **ventaja de las pruebas de caja negra** es que son independientes del lenguaje o paradigma de programación utilizado, por lo que son válidas tanto para programación estructurada como para programación orientada a objetos.

VÍDEO: CLASES DE EQUIVALENCIA Y VALORES LÍMITE [ENLACE]



PRUEBAS DE CAJA NEGRA

Clases de equivalencia

Se deben diseñar los casos de prueba de forma que prueben la mayor funcionalidad posible del programa, pero que no incluyan demasiados valores. ¿Por dónde empezar? ¿Qué valores se deben escoger? tendremos que seguir tres pasos:

1. Identificar las condiciones de las entradas y las salidas.
2. Identificar, a partir de las restricciones, las clases de equivalencia de las entradas y las salidas. Por cada condición de entrada se identifican clases de equivalencia válidas y no válidas. Existen un conjunto de criterios/reglas heuristicas que ayudan a su identificación:

PRUEBAS DE CAJA NEGRA

Clases de equivalencia

- Si una condición de entrada especifica un rango de valores para los datos de entrada, por ejemplo, si se admite del 10 al 50:
  - Se creará una clase válida ($10 \leq X \leq 50$) y dos clases no válidas, una para los valores superiores ($X > 50$) y la otra para los inferiores ($X < 10$).
- Si una condición de entrada especifica un valor válido para los datos de entrada, por ejemplo, si la entrada comienza con mayúscula:
  - Se crea una clase válida con el valor válido (con la primera letra mayúscula) y otra de no válida para el resto (con la primera letra minúscula).

PRUEBAS DE CAJA NEGRA

Clases de equivalencia

- Si se especifica un número de valores de entrada, por ejemplo, si se han de introducir tres números seguidos:
  → Se creará una clase válida (con tres valores) y dos de no válidas (una con menos de dos valores y otra con más de tres valores).

- Si hay un conjunto de datos de entrada concretos válidos, se generará una clase para cada valor válido, por ejemplo, si la entrada debe ser rojo, naranja o verde:
  → Se generarán tres clases válidas y otra por un valor no válido (por ejemplo, azul).

- Si no se han recogido ya todas las casuísticas con las clases anteriores, se debe seleccionar una clase para cada posible clase de resultado.

PRUEBAS DE CAJA NEGRA

Clases de equivalencia

3. Identificar los casos de prueba a partir de las clases de equivalencia detectadas. Para ello se deben seguir los siguientes pasos:

- Elegir un valor que represente cada clase de equivalencia (valores límite).
- Diseñar casos de prueba que incluyan los valores de todas las clases de equivalencia identificadas.

PRUEBAS DE CAJA NEGRA

Ejemplo, se quieren definir las pruebas de caja negra para una función que devuelve el nombre del mes a partir de su valor numérico:

```
String nombre;
nombre = NomDelMes (3);
El valor del nombre será Marzo.
```

¿Cuales serían sus clases de equivalencia?

Serían un total de 3 clases de equivalencia:

1 <= X <= 12
X > 12
X < 1

PRUEBAS DE CAJA NEGRA

Resumen Clases de equivalencia

- 1º. Identificar las condiciones de las entradas y las salidas.
- 2º. Identificar las clases de equivalencia válidas y no válidas.
- 3º. Identificar los casos de prueba (Escenario – Resultado Esperado)

Veamos a continuación un ejemplo práctico.

EJEMPLO

Construcción de una batería de pruebas para detectar posibles errores en la construcción de los identificadores/variables de un hipotético lenguaje de programación. Las reglas que determinan sus construcción sintáctica son:

- No debe tener mas de 15 ni menos de 5 caracteres.
- El juego de caracteres utilizables es:

– Letras (Mayúsculas y minúsculas)

– Dígitos (0,9)

– Guion (-)

- Se distinguen las mayúsculas de las minúsculas.
- El guion no puede estar ni al principio ni al final, pero puede haber varios consecutivos.
- Debe contener al menos un carácter alfabético.
- No puede ser una de las palabras reservadas del lenguaje.

EJEMPLO

Condiciones de Entrada	Clases de equivalencia válida	Clases de equivalencia no válida
-Entre 5 y 15 caracteres		
-El identificador debe estar formado por letras, dígitos y guión		
-Se diferencia entre letras mayúsculas y minúsculas		
-El guión no puede estar al principio, ni al final -Puede haber varios seguidos en el medio		
-Debe contener al menos un carácter alfabético		
-No se pueden usar palabras reservadas		

EJEMPLO

Realizamos este cuadro a partir de los criterios estudiados en las diapositivas 10 y 11 para identificar las clases de equivalencia válidas y no válidas.

Condiciones de Entrada	Clases de equivalencia válida	Clases de equivalencia no válida
-Entre 5 y 15 caracteres	**1**. 5≤nº caracteres Ident. ≤15	**2**. nº caracteres Id<5
		3. 15< nº caracteres
-El identificador debe estar formado por letras, dígitos y guión	**4.** Todos los caracteres del Ident. ∈{letras, dígitos, guión}	**5.** Alguno de los caracteres del Ident. ∉{letras, dígitos, guión}
-Se diferencia entre letras mayúsculas y minúsculas	**6.** Palabra declarada ∈{Identificadores válidos}	**7.** Utilizar la misma palabra con alguna letra conmutada para hacer referencia al mismo identificador
-El guión no puede estar al principio, ni al final -Puede haber varios seguidos en el medio	**8.** Identificador sin guiones en los extremos y con varios consecutivos en el medio	**9**. Identificador con guión al principio
		10. Identificador con guión al final
-Debe contener al menos un carácter alfabético	**11.** Al menos un carácter del Ident. ∈{letras}	**12.** Ningún carácter del Ident. ∈{letras}
-No se pueden usar palabras reservadas	**13.** El Identificador ∉{palabras reservadas}	**14, 15, 16** ....un caso por cada palabra reservada

EJEMPLO

Identificamos ahora los
casos de pruebas.

Identificador	Clases de equivalencia cubiertas	Resultado
Num-1---d3 (10)	1,4,6,8,11,13 (todas las válidas)	El sistema acepta el identificador
Nd3	2	Mensaje de error
Num-1-letra3---d3 (17)	3	Mensaje de error
Nu%m-1---d3 (11)	5	Mensaje de error
NuM-1---d3 (10)	7	Mensaje de error
-um-1---d3 (10)	9	Mensaje de error
Num-1---d- (10)	10	Mensaje de error
456-1---23 (10)	12	Mensaje de error
Real	14	Mensaje de error
..(el resto de palabras reservadas)..	15,16.....	Mensaje de error

PRUEBAS DE CAJA NEGRA

Análisis de valores límite

La experiencia muestra que los casos de prueba que exploran las condiciones límite producen mejor resultado que aquellos que no lo hacen.

Las condiciones límite son aquellas que se hayan en los márgenes (por encima y por debajo) de las clases de equivalencia, tanto de entrada como de salida.

Por lo tanto, el análisis de valores límite complementa la técnica de las clases de equivalencia de manera que:

Al seleccionar el elemento representativo de la clase de equivalencia, en lugar de coger uno cualquiera, se escogen los valores al límite y, si se considera oportuno, un valor intermedio. Además, también se intenta que los valores en la entrada provoquen valores límite a los resultados.

PRUEBAS DE CAJA NEGRA

Análisis de valores límite

A la hora de escoger los representantes de cada clase se seguirán las siguientes recomendaciones:

1. Rango de valores como condición de entrada	**1** caso que ejercite el valor máximo del rango
	1 caso que ejercite el valor mínimo del rango
	1 caso que ejercite el valor justo por encima del máximo del rango
	1 caso que ejercite el valor justo por debajo del mínimo del rango
2. Valor numérico específico como condición de entrada	**1** caso que ejercite el valor numérico específico
	1 caso que ejercite el valor justo por encima del valor numérico específico
	1 caso que ejercite el valor justo por debajo de valor numérico específico
3. Rango de valores como condición de salida	Generar casos de prueba según el criterio 1 que ejerciten dichas condiciones de salida
4. Valor numérico específico como condición de salida	Generar casos de prueba según el criterio 2 que ejerciten dichas condiciones de salida
5. Estructura de datos como condición de salida o de entrada	**1** caso que ejercite el primer elemento de la estructura
	1 caso que ejercite el último elemento de la estructura

EJEMPLO

Ejemplo: número entero x

0<=X <=100

5 clases de equivalencia:

1. X <0 No válida
2. 0<= X <=100 Válida
3. X >100 No válida
4. X decimal No válida
5. X alfanumérico No válida

Análisis de valores límite

{ 0,100,1,99,-1,101}

Casos de prueba

N°	Caso de prueba		Representante
1	X <0	No valido	-1
2	0<= X <=100	Válido	0, 1, 99, 100
3	X >100	No valido	101
4	X decimal	No valido	1,5
5	X alfanumérico	No valido	a

ACTIVIDAD

- Aplicación bancaria. Datos de entrada:
  - **Código de área:** número de 3 dígitos que no empieza por 0 ni por 1
  - **Nombre de identificación de operación**: 6 caracteres
  - **Órdenes posibles**: “cheque”, “depósito”, “pago factura”, “retirada de fondos”

ACTIVIDAD

Parámetro de entrada	Regla heurística a aplicar	Clases válidas	Clases inválidas
Código de área	Condición booleana (¿es un número?) + rango valores ([200..999])	**1.** 200<=código<=999	**2.** código < 200 **3.** código > 999 **4.** no es número
Clave identificativa de la operación	Conjunto finito de valores (¿son 6 caracteres?)	**5.** 6 caracteres exactos	**6.** menos de 6 caracteres **7.** más de 6 caracteres
Órdenes posibles	Conjunto de valores admitidos (¿orden válida?)	**8.** "cheque" **9.** "depósito" **10.** "pago factura" **11.** "retirada de fondos"	**12.** no es orden válida

ACTIVIDAD

Diseño de **casos de prueba** para cubrir tantas **clases de equivalencia válidas** como sea posible **(1, 5, 8, 9, 10 y 11)**

Código de área	Clave identificativa	Orden	Clases válidas cubiertas
300	Nómina	"cheque"	1, 5, 8
400	Viajes	"depósito"	1, 5, 9
500	Coches	"pago factura"	1, 5, 10
600	Comida	"retirada de fondos"	1, 5, 11

Diseño de **casos de prueba** para cubrir **una y solo** una **clase de equivalencia inválida** cada vez **(2, 3, 4, 6, 7 y 12)**

Código de área	Clave identificativa	Orden	Clases inválidas cubiertas
150	Viajes	"pago factura"	2, 5, 10
1100	Coches	"depósito"	3, 5, 9
#Ab20	Comida	"retirada de fondos"	4, 5, 11
220	Luz	"depósito"	1, 6, 9
400	Teléfono	"cheque"	1, 7, 8
500	Viajes	"pagos varios&&"	1, 5, 12

ACTIVIDAD

Extendemos los casos de prueba para **trabajar** con los **valores límite**

Parámetro de entrada	Regla heurística a aplicar	Clases válidas	Clases inválidas
Código de área	Condición booleana (¿es un número?) + rango valores ([200..999])	**13.** código=200 **14.** código=201 **15.** código=998 **16.** código=999	**17.** código=199 **18.** código=1000
Clave identificativa de la operación	Conjunto finito de valores (¿son 6 caracteres?)	No hay cambios	**19.** 5 caracteres **20.** 7 caracteres
Órdenes posibles	Conjunto de valores admitidos (¿orden válida?)	No hay cambios	No hay cambios

A continuación generaríamos los **casos de prueba** para cubrir las **nuevas clases** (13..20)

ACTIVIDAD

Determinar las clases de equivalencia para las siguientes condiciones de entrada

Un usuario puede conectarse al banco por Internet y realizar una serie de operaciones bancarias. Una vez accedido al banco con las consiguientes medidas de seguridad (clave de acceso y demás), se requiere la siguiente entrada:

- **Código del banco**. En blanco o número de tres dígitos. En este último caso, el primero de los tres tiene que ser mayor o igual que 1
- **Código de sucursal**. Un número de cuatro dígitos. El primero de ellos mayor de 0
- **Número de cuenta.** Número de cinco dígitos
- **Clave personal**. Valor alfanumérico de cinco posiciones. Este valor se introducirá según la orden que se desee realizar
- **Orden**. Puede estar en blanco o ser una de las dos cadenas siguientes:
  - "Talonario"
  - "Movimientos"

  En el primer caso el usuario recibirá un talonario de cheques, mientras que en el segundo recibirá los movimientos del mes en curso. Si este código está en blanco, el usuario recibirá los dos documentos

RESUMEN CLASES DE EQUIVALENCIA Y VALORES LÍMITE

Gracias a estas técnicas podemos garantizar que:

- Todas las clases de equivalencia válidas e inválidas tienen unos valores lo suficientemente representativos de cada clase, qué, además explotan los límites operacionales.
- Comprobamos si los resultados coinciden con los esperados.

Aún así, no podemos garantizar que el programa esté 100% libre de errores.

PRUEBAS DE CAJA NEGRA

Errores típicos

Para diseñar casos de prueba, también se puede aprovechar la **experiencia previa.** Hay una serie de errores que se repiten mucho en los programas, y podría ser una buena estrategia utilizar casos de prueba que se centren en buscar estos errores. De este modo, se mejorará la elección de los representantes de las clases de equivalencia:

- El valor **cero** suele provocar errores, por ejemplo, una división por cero aborta el programa. Si se tiene la posibilidad de introducir ceros a la entrada, se debe escoger en los casos de prueba.

PRUEBAS DE CAJA NEGRA

- Cuando se debe introducir una **lista de valores**, habrá que centrarse en la posibilidad de no introducir **ningún valor**, o introducir **uno.**
- Hay que pensar que el **usuario puede introducir entradas que no son normales**, por eso es recomendable ponerse en el peor caso.
- Los **desbordamientos de memoria** son habituales, por eso se debe intentar introducir **valores tan grandes como sea posible.**

PRUEBAS DE CAJA NEGRA

Manejo de interfaz gráfica

No sólo se debe hablar de entradas de textos, también hay que tener en cuenta los **entornos gráficos** donde se llevan a cabo las entradas de valores o donde se visualizan los resultados. Actualmente, la mayoría de programas suelen interactuar con el usuario haciendo uso de sistemas gráficos que cada vez son más complejos, con lo cual se pueden generar errores.

PRUEBAS DE CAJA NEGRA

Manejo de interfaz gráfica

Las **pruebas** de interfaz gráfica de usuario deben incluir:

- Pruebas sobre menús y uso de ratón.
- Pruebas sobre ventanas: iconos de cerrar, minimizar…
- Pruebas de entrada de datos: cuadro de textos, listas desplegables…
- Pruebas de documentación y ayuda del programa.
- Otros.

RESUMEN

- El proceso de prueba consiste en ejecutar el programa con el fin de localizar errores.
- La prueba es una actividad incompleta.
- **Propósito** de las técnicas de **diseño de casos de prueba** es **reducir el número de casos de prueba sin mermar la efectividad de la prueba.**
- Existen dos enfoques a la hora de abordar el diseño de los casos de prueba:
  - **Caja Negra:** evalúa la funcionalidad del sistema (lo que hace).
  - **Caja Blanca:** evalúa la lógica interna (cómo lo hace).

Thats All