



# Pruebas de Software



# Diseño de Pruebas de Software

# Agenda

- Propósito de diseño de Pruebas
- Insumos para el proceso de Diseño de Pruebas
- Definición caso de prueba
- Estructura de un caso de prueba
- Proceso General del diseño de Pruebas
  - Identificación del alcance proceso de pruebas
  - Definición de escenarios – Técnicas Diseño de Pruebas
  - Especificación de casos de prueba

# Propósito de Diseño de Pruebas

El propósito fundamental de esta fase es **generar instrumentos de prueba:**

---

***Eficientes***, es decir, construir la cantidad de casos de prueba necesarios que permitan identificar si el software desarrollado, cumple con los requerimientos contratados por el Cliente.

---

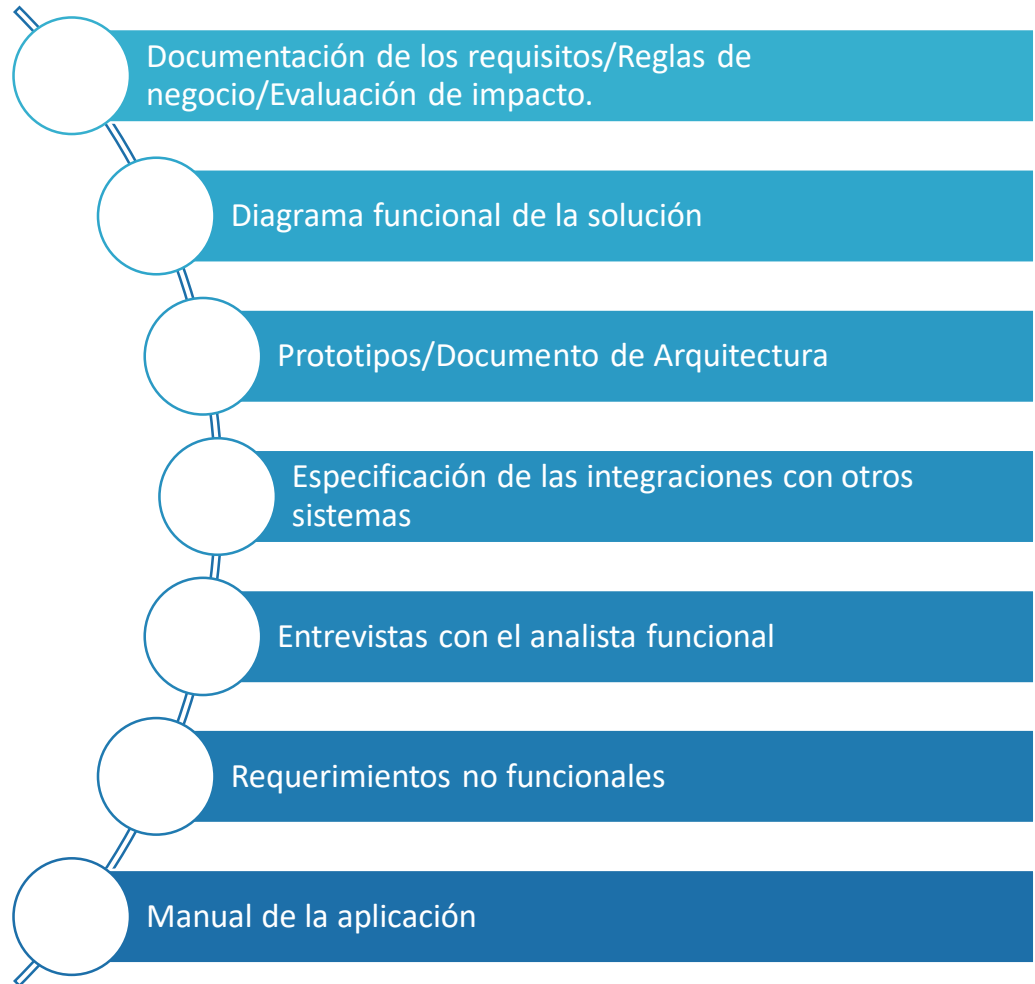
***Confiables***, es decir, que cubran completamente el alcance de las pruebas y que su ejecución tenga como objetivo fundamental la identificación de no conformidades en el software.

---

***Repetibles***, es decir, instrumentos de pruebas documentados que permita reutilizar las condiciones de pruebas para nuevas versiones de productos, logrando así mayor eficiencia en el proceso de pruebas de software.

---

# Insumos para el Proceso de Diseño de Pruebas

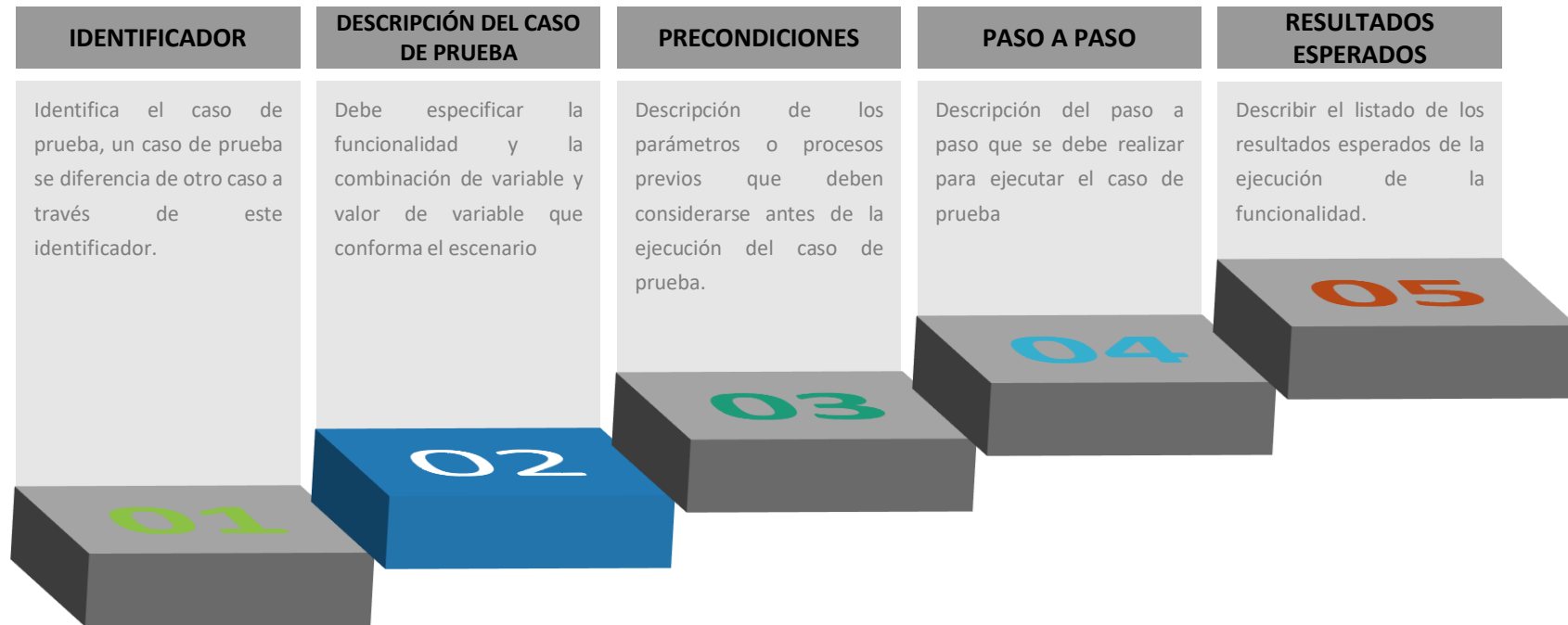


# Definición Caso de Prueba

Definición caso de prueba:

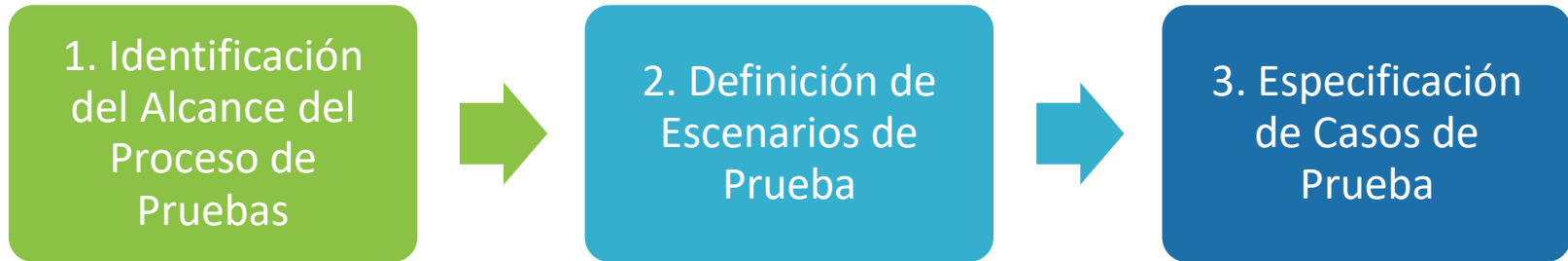
- Un **caso de prueba** o **test case** es un **conjunto de condiciones o variables** bajo las cuales un analista **determinará** si una **aplicación**, un sistema de software (software system), o una característica de éstos es parcial o completamente satisfactoria.
- **Caso de Prueba:** Un caso de prueba es una serie de condiciones de ejecución y resultados esperados, desarrollados para verificar el cumplimiento de requisitos de un sistema.

# Estructura de un caso de prueba





# Proceso General de Diseño de Pruebas



# 1. Identificación del Alcance del Proceso de Pruebas

# Proceso de Diseño de Pruebas

## 1. Identificación del Alcance del Proceso de Pruebas

Se debe identificar las funcionalidades que hacen parte de la solución de software que deben ser probadas.

Se descompone la aplicación en procesos, subprocesos y funcionalidades.

**Nota: para los productos en Mantenimiento se debe Identificar las funcionalidades impactadas por la nueva funcionalidad o ajuste.**

# 1. Identificación del Alcance del Proceso de Pruebas

Proceso	Subproceso	Funcionalidad
Registro de usuario	Registro de usuario	Registrar usuario
		Validar Código SMS
Manejo de Sesión	Manejo de Sesión	Iniciar sesión
		Recuperar contraseña
		Cierre de sesión
Gestión Cuenta de usuario	Editar Perfil	Editar perfil
	Administrar direcciones	Agregar dirección
		Eliminar dirección
		Editar dirección
	Cambio de contraseña	Cambiar contraseña
	Administrar tarjetas de crédito	Agregar Tarjeta de Crédito
		Marcar Tarjeta de Crédito Favorita
		Eliminar Tarjeta
	Gestión cuenta LifeMiles	Vincular Cuenta LifeMiles
		Desvincular Cuenta LifeMiles
		Modificar Cuenta LifeMiles
	Contáctenos	Contáctenos
Gestión Cupón	Gestión Cupón	Redimir Cupón
Consultar Productos	Consultar Productos	Búsqueda de productos
		Consulta detalle de producto
		Consultar productos por Categorías
		Consultar Carrusel Te puede interesar
		Consulta carrusel Destacado
		Consultar Carrusel Recomendados para ti
Gestión carro compras	Gestión carro compras	Consultar de pedidos anteriores
		Agregar productos
		Consultar carrito
		Modificar productos en carrito
		Eliminar producto del carrito
Gestión del pedido	Gestión del pedido	Eliminar todo el pedido
		Programar Pedido
		Visualizar resumen del pedido
		Ingresar cupón
		Acumulación de millas LifeMiles
		Agregar dirección de domicilio
		Seleccionar método de pago
		Agregar comentarios
		Pagar Pedido
Gestión Cuenta	Visualizar pedidos	Calificar pedidos
Chat	Chat	Chat

## 2. Definición de Escenarios de Prueba

# Proceso de Diseño de Pruebas

## 2. Definición de Escenarios de Pruebas por Funcionalidad

Es importante para cada producto de software tener la línea base de casos de prueba.

### 2.1 Identificar Variables/Condiciones y Valores por Funcionalidad

Para cada una de las funcionalidades es necesario identificar las situaciones o condiciones de operación que se pueden presentar y que son susceptibles a una verificación previa.

¿cómo identificar variables/Condiciones de una Funcionalidad?  **Conocimiento Reglas de Negocio**

¿Cómo identificar los valores de las variables/condiciones?  **Técnicas de Diseño de Pruebas**

# Proceso de Diseño de Pruebas

Una **Variable o Condición de prueba** es un aspecto relevante de la funcionalidad y que dependiendo del valor que esta variable adquiera, el funcionamiento del sistema es diferente.


**Ejemplos:**

## Facturación de un servicio público

Variable/condición de prueba



Valores de variable



Estado del Producto	Estado Financiero	Estrato Económico	Categoría
Activo	Al día	1	Residencial
Suspendido	Con deuda	2	Comercial
Con orden de suspensión	Mora	3	Industrial
		4	
		5	
		6	

# Proceso de Diseño de Pruebas

## Registro de Afiliado EPS

Variable/condición de prueba	→					
Valores de variable	⌈	Tipo de Afiliado	Edad	Regimen	Nivel de Sisben	Afiliado a otra EPS?
		Cotizante	Menor de edad	Contributivo	1	Si
		Beneficiario	Mayor de edad	Subsidiado	2	No
					3	

Las variables identificadas siempre deben mantener la secuencia de interacción, es decir, su orden de ingreso para la operación del software.

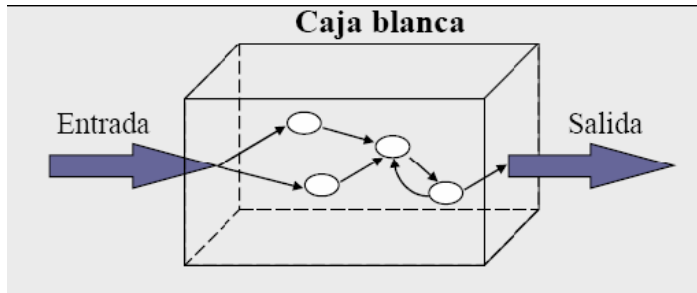


# Técnicas de Diseño de prueba



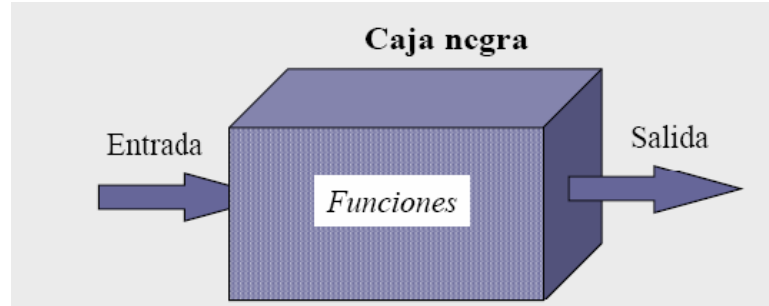
- La aplicación de técnicas de diseño de pruebas, permite identificar los casos de prueba necesarios para generar confiabilidad en la evaluación de las funcionalidades del producto de Software.

**Técnicas de caja blanca o estructurales**, que se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa.



# Técnicas de Diseño de prueba

**Técnicas de Caja Negra o Funcionales.** También conocidas como **Pruebas de Comportamiento**, estas pruebas se basan en la *especificación* del programa o componente a ser probado para elaborar los casos de prueba. El componente se ve como una “Caja Negra” cuyo comportamiento sólo puede ser determinado estudiando sus entradas y las salidas obtenidas a partir de ellas.



# Técnicas de Diseño de prueba

## Porqué usar Técnicas de Diseño de Prueba? / Ventajas

- **Imposibilidad de Prueba Exhaustiva del software.** Las técnicas de diseño de pruebas nos ayuda a determinar los criterios para seleccionar los casos de prueba.
- **Garantizar el mayor grado de confianza** posible en que se detectaran los defectos del software.
- **Equilibrio entre recursos y garantía** para descubrir los defectos existentes.



# Técnicas de Diseño de prueba Caja Negra

- ✓ **El tester observa el objeto de prueba como una caja negra.**
  - La estructura interna del objeto de prueba es irrelevante o desconocida.
- ✓ **Los casos de prueba se obtienen a partir del análisis de la especificación (funcional y no funcional) de un componente o sistema.**
  - Prueba del comportamiento entrada / salida (input / output).
- ✓ **La funcionalidad es el foco de atención!**
  - La técnica de caja negra también **se denomina prueba funcional o prueba orientada a la especificación.**

# Técnicas de Diseño de prueba Caja Negra

❖ Partición de Equivalencia

❖ Análisis de valor Limite

❖ Tabla de Decisión

❖ Arreglos Ortogonales

## **Técnicas primer nivel**

**Foco:** Identificar valores de las variables/condiciones

## **Técnicas segundo nivel**

**Foco:** Identificar escenarios de prueba



# Partición de Equivalencia

## Definición:

Se basa en la **división del campo de entrada** en un conjunto de clases de datos denominadas **clases de equivalencia**

Conjunto de Clases Equivalentes → Conjunto de datos que definen entradas Válidas y No Válidas al Sistema

Entradas Válidas:

- Generan un valor esperado

Entradas No Válidas:

- Generan un valor "inesperado"

Una clase de equivalencia representa un conjunto de estados válidos o no válidos para una condición de entrada. Todos los valores para los cuales se espera que el programa tenga un comportamiento común se agrupan en una CE.

Típicamente, una condición de entrada es **un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica.**

# Criterios de Identificación de Clases de Equivalencia

- Por cada Variable/condición de entrada se identifican clases de equivalencia válidas y no válidas.
- Sin embargo existen un conjunto de criterios que ayudan a su identificación.

Condiciones de Entrada	Número de clases de equivalencia válida	Número de clases de equivalencia no válida
1. Rango de valores	<b>1</b> clase que contemple los valores del rango	<b>2</b> clases fuera del rango, una por encima y otra por debajo de éste
2. Valor específico	<b>1</b> clase que contemple dicho valor	<b>2</b> clases que representen un valor por encima y otro por debajo
3. Elementos de un conjunto tratados de forma diferente por el programa	<b>1</b> clase de equivalencia por cada elemento	<b>1</b> clase que represente un elemento fuera del conjunto
4. Condición lógica	<b>1</b> clase que cumpla la condición	<b>1</b> clase que no cumpla la condición

# Partición de Equivalencia

Divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba.

1. Si una condición de entrada especifica un rango, se define una clase de equivalencia válida y dos inválidas.

<b>Técnica:</b>	Partición Equivalente
<b>T. Entrada:</b>	Rango
<b>Proceso:</b>	Registro de Venta
<b>Variable:</b>	Descuento
<b>Valores:</b>	Entre 0 - 100%
<b>Clases de equivalencia:</b>	0-100% Válida
	< 0% No Válida
	> 100% No Válida



# Partición de Equivalencia

2. Si una condición de entrada requiere un valor específico, se define una clase de equivalencia válida y dos inválidas

<b>Técnica:</b>	Partición Equivalente	
<b>T. Entrada:</b>	Valor Especifico	
<b>Proceso:</b>	Afiliación usuario a Entidad Promotora de Salud -EPS	
<b>Variable:</b>	Tipo Documento de Identidad	
<b>Valores:</b>	Cedula Ciudadanía	
<b>Clases:</b>	Cedula Ciudadanía	Válida
	Tarjeta Identidad	No Válida
	Registro Civil	No Válida

3. Si se especifica un conjunto de valores admitidos y el programa trata de forma distinta a cada uno de ellos , se crea una clase valida por cada valor, y una no valida.

Ej: Tres tipos de inmuebles :

**Validas:** Apartamento, Chalets, Local comercial

**No valida:** Automóvil

# Partición de Equivalencia

4. Si una condición de entrada es lógica, se define una clase válida y una inválida.

<b>Técnica:</b>	Partición Equivalente
<b>T. Entrada:</b>	Variable Lógica
<b>Proceso:</b>	Afiliación usuario a Entidad Promotora de Salud -EPS
<b>Variable:</b>	Registro en otra EPS
<b>Valores:</b>	Falso
<b>Clases:</b>	Verdadero    No Válida
	Falso        Válida

# Partición de Equivalencia – Método Aplicación

## Proceso:

1. Identifique las variables
2. Defina las clases de equivalencia (CE) válidas, no válidas y sus representantes para cada una de las condiciones/variables.
3. Genere las combinaciones para casos de prueba con CE válidas.
4. Genere las combinaciones para casos de prueba con CE no válidas.
5. Diseñe cada caso de prueba completo.

# Ejemplo clases de Equivalencia

Un programa espera un número entero mayor a 0 en donde el programa realiza:

- Si el número es par, la función devolverá “Es un número par!”
- Si el número es impar, la función devolverá “Es un número impar!”

# Ejemplo clases de Equivalencia

## **Proceso:**

1. Identifique las variables

# Ejemplo clases de Equivalencia

Variables	Clases de equivalencia	Tipo	Representante
Valor			

# Ejemplo clases de Equivalencia

## Proceso:

1. Identifique las variables
2. Defina las clases de equivalencia (CE) válidas, no válidas y sus representantes.

# Ejemplo clases de Equivalencia

Variables	Clases de equivalencia	Tipo	Representante
Valor	Par mayor 0	Válida	10
	Impar mayor 0	Válida	5
	Decimal	No válida	1,5
	Entero menor o igual a 0	No válida	-3
	No numérico	No válida	abc



# Ejemplo clases de Equivalencia

## Proceso:

1. Identifique las variables
2. Defina las clases de equivalencia (CE) válidas, no válidas y sus representantes.
3. Genere las combinaciones para casos de prueba con CE válidas.

# Ejemplo clases de Equivalencia

Variables	Clases de equivalencia	Representante	Tipo	T01	T02	T03	T04	T05
Valor	Par mayor 0	10	Válida	X				
	Impar mayor 0	5	Válida		X			
	Decimal	1,5	No válida					
	Entero menor o igual a 0	-3	No válida					
	No numérico	abc	No válida					

# Ejemplo clases de Equivalencia

## Proceso:

1. Identifique las variables
2. Defina las clases de equivalencia (CE) válidas, no válidas y sus representantes.
3. Genere las combinaciones para casos de prueba con CE válidas.
4. Genere las combinaciones para casos de prueba con CE no válidas.

# Ejemplo clases de Equivalencia

Variables	Clases de equivalencia	Representante	Tipo	T01	T02	T03	T04	T05
Valor	Par mayor 0	10	Válida	X				
	Impar mayor 0	5	Válida		X			
	Decimal	1,5	No válida			X		
	Entero menor o igual a 0	-3	No válida				X	
	No numérico	abc	No válida					X

# Ejemplo clases de Equivalencia

Por parte del código de un programa trata el precio final de un artículo con base a su precio de venta al público, un descuento en % y el precio del porte (6, 9 ó 10 euros, dependiendo del tipo de porte).

Variable	Clase de equivalencia	Estado	Representante
Precio	EC <sub>11</sub> : $x \geq 0$	Válido	1000,00
	EC <sub>12</sub> : $x < 0$	No válido	-1000,00
	EC <sub>13</sub> : x valor no numérico	No válido	fred
Descuento	EC <sub>21</sub> : $0\% \leq x < 100\%$	Válido	10%
	EC <sub>22</sub> : $x < 0\%$	No válido	-10%
	EC <sub>23</sub> : $x > 100\%$	No válido	200%
	EC <sub>24</sub> : x valor no numérico	No válido	fred
Precio del porte	EC <sub>31</sub> : $x = 6$	Válido	6
	EC <sub>32</sub> : $x = 9$	Válido	9
	EC <sub>33</sub> : $x = 12$	Válido	12
	EC <sub>34</sub> : $x \notin \{6, 9, 12\}$	No válido	4
	EC <sub>35</sub> : x valor no numérico	No válido	fred

## Suposiciones:

- El precio de venta al público de un artículo está dado por un número con dos decimales.
- El descuento es el valor porcentual sin decimales entre 0% y 100%.
- El precio del porte puede ser 6, 9 ó 12.

# Ejemplo clases de Equivalencia

Se obtienen 10 casos de prueba: 3 casos de prueba positivos (valores válidos) y 7 casos de prueba negativos (valores no válidos).

Variable	Estado	Representante	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10
Precio	Válido	1000,00	*	*	*			*	*	*	*	*
	No válido	-1000,00				*						
	No válido	fred					*					
Descuento	Válido	10%	*	*	*	*	*				*	*
	No válido	-10%						*				
	No válido	200%							*			
	No válido	fred								*		
Precio del porte	Válido	6	*			*	*	*	*	*		
	Válido	9		*								
	Válido	12			*							
	No válido	4									*	
	No válido	fred										*

# Ejercicio

El nuevo proyecto Banca Móvil ha creado la cuenta tipo 'BM' que no permite retiro con talonario. Por lo anterior, la funcionalidad de retiro de cuenta con talonario ha sido impactada.

Adicionalmente, debe tenerse en cuenta que:

El monto máximo de retiro diario con talonario es 100.000.

# Ejercicio

Los tipos de cuenta existentes son:

- Cuenta de ahorros
- Cuenta corriente
- Cuenta Banca Móvil (BM) *(nuevo)*

Toda cuenta puede tener los siguientes estados: Activo, Inactivo, Embargada, Saldada.

Sólo es posible realizar retiros con talonario a cuentas de ahorro activas.



# Ejercicio

1. Identifique las variables:

- Tipo de cuenta
- Estado de cuenta
- Monto de retiro

# Ejercicio

2. Defina las clases de equivalencia (CE) válidas, no válidas y sus representantes.

Variable	Clase de equivalencia	Estado	Representante
Tipo de cuenta	Cuenta de ahorros	Válido	Ahorros
	Cuenta corriente	No Válido	Corriente
	Banca Móvil	No Válido	BM
	Tercero	No válido	Tercero
Estado de cuenta	Activa	Válido	A
	Inactiva	No Válido	I
	Embargada	No Válido	E
	Saldada	No Válido	S
	EC != (A, I, E, S)	No válido	F
Monto retiro	$0 < X \leq 100.000$	Válido	50.000
	$> 100.000$	No válido	200.000
	$< 0$	No válido	-5.000

# Ejercicio

3. Genere las combinaciones para casos de prueba con CE válidas.

Variable	Clase de equivalencia	Estado	Representante	T01
Tipo de cuenta	Cuenta de ahorros	Válido	Ahorros	X
	Cuenta corriente	No válido	Corriente	
	Banca Móvil	No válido	BM	
	Tercero	No válido	Tercero	
Estado de cuenta	Activa	Válido	A	X
	Inactiva	No Válido	I	
	Embargada	No Válido	E	
	Saldada	No Válido	S	
	EC != (A, I, E, S)	No válido	F	
Monto retiro	$1 \leq X \leq 100.000$	Válido	50.000	X
	$> 100.000$	No válido	200.000	
	$< 1$	No válido	-5.000	

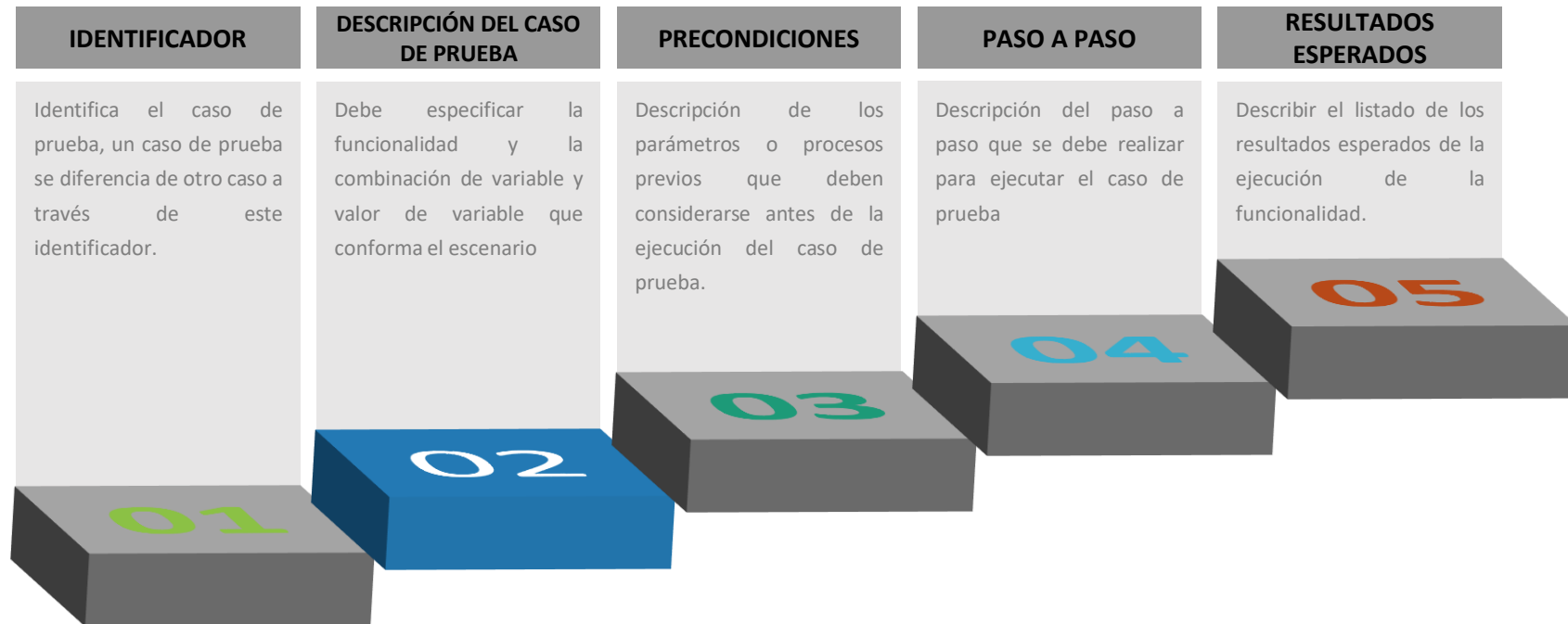


# Ejercicio

## Proceso:

1. Identifique las variables
2. Defina las clases de equivalencia (CE) válidas, no válidas y sus representantes.
3. Genere las combinaciones para casos de prueba con CE válidas.
4. Genere las combinaciones para casos de prueba con CE no válidas.
5. Diseñe cada caso de prueba completo.

# Estructura de un caso de prueba



# Ejercicio

5. Diseñe cada caso de prueba completo.

Identificador	Descripción del caso de prueba	Precondiciones	Paso a Paso	Resultados esperados
T01	Realizar un retiro: Tipo de cuenta: ahorros Estado de la cuenta: Activa Monto de Retiro: \$ 50.000	La cuenta debe tener saldo superior o igual a \$50.000	.....	Verificar:  - El retiro de la cuenta es exitoso. - Verificar que el saldo actual es igual a: Saldo anterior - Monto de retiro.

# Ejercicio Práctico

## Ejercicio:

- Realice los ejercicios propuestos en el taller
- Tiempo estimado 15 minutos





# Análisis de Valor Límite

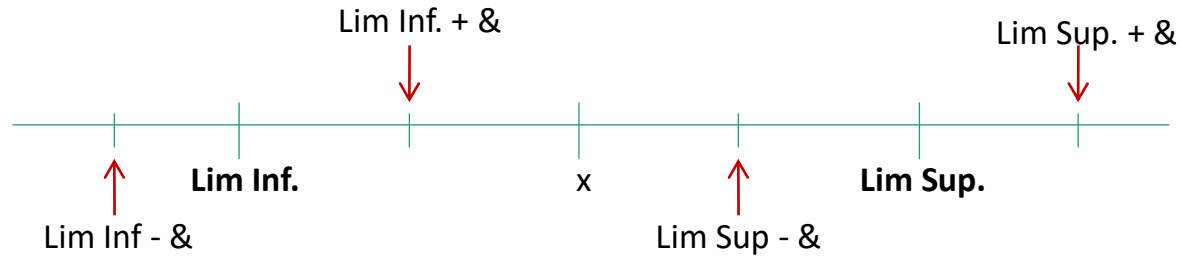


Los **valores frontera** (valores límite) de la clase de equivalencia deben ser probados de **forma intensiva**.

- ¿Porqué prestar más atención a los límites?
  - Frecuentemente los límites del rango de valores no están bien definidos o conducen a distintas interpretaciones.
  - Comprobar si los límites han sido implementados (programados) correctamente
- Importante:
  - *¡La experiencia demuestra que con mucha frecuencia, los errores tienen lugar en los límites del rango de valores!*

# Análisis de Valor Límite

- El análisis de valores límite amplía la técnica de partición en clases de equivalencia **introduciendo una regla para seleccionar a los representantes.**



$\&$  es el menor Incremento definido por el valor.

# Análisis de Valor Límite

- Si un Campo de Entrada está definido como un único valor numérico, por ejemplo,  $x = 5$ , los valores correspondientes al entorno también serán utilizados.
  - Los representantes (de la clase y su entorno) son:

4 , 5 y 6

# Análisis de Valor Límite

- **Ejemplo a:**
  - Rango de valores para un descuento en %:  $0.00 \leq X \leq 100.00$
- **Definición de CE 3 clases:**
  - CE:  $X < 0$  ( No valido)
  - CE:  $0.00 \leq X \leq 100.00$  (Valido)
  - CE:  $X > 100$  ( No valido)
- **Análisis de valores límite,**
  - Extiende los representantes a:
    - 2. CE: -0.01; 0.00; 00.1; 99.99; 100.00; 100.01

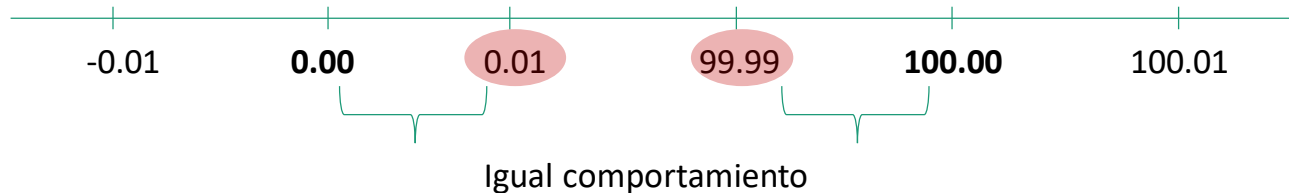


# Análisis de Valor Límite

## Punto de vista alternativo:

- **Ejemplo b:**

- Rango de valores para un descuento en %:  $0.00 \leq X \leq 100.00$
- **CE válido:**  $0.00 \leq X \leq 100.00$
- **Análisis de valores límite.**
  - Los representantes adicionales son; -0.01; 0,00; 100.00; 100.01
    - 0,01 - mismo comportamiento que 0.00
    - 99.99 - mismo comportamiento que 100.00



# Análisis de Valor Límite

## **Pasos simples para usar la técnica**

1. Identificación de las clases de equivalencia.
2. Identificación de los valores límites de cada equivalencia (clases de equivalencia válidas).
3. Crear casos de prueba.

## **Ventajas de la técnica:**

1. La técnica reduce el número de casos de pruebas que deben ser creados y ejecutados.
2. Esta técnica permite elegir un subconjunto de las pruebas que son eficiente y eficaces en encontrar no conformidades

# Ejercicio

[illegible]

# Ejercicio Práctico

## **Ejercicio:**

- Realice los ejercicios propuestos en el taller
- Tiempo estimado 15 minutos
- Solución en grupo al finalizar el tiempo estimado.





# Tablas de Decisión

- ✓ Las tablas de decisión representa la relación lógica entre **condiciones** (entradas) y **acciones**(salidas) , semejante a una relación causa efecto.
- ✓ Son utilizadas para registrar reglas del negocio complejas basadas en un conjunto de condiciones y acciones, las cuales deben ser implementadas en el sistema.

Condiciones	Reglas							
Condición 1	S	S	S	S	N	N	N	N
Condición 2	S	S	N	N	S	S	N	N
Condición 3	S	N	S	N	S	N	S	N
Acción 1	X	X						
Acción 2				X		X		X
Acción 3			X				X	
Acción 4					X			

**Los casos de prueba son derivados considerando combinaciones entre las condiciones y las acciones.**

# Tablas de Decisión

Cada regla es un  
caso de prueba

	Regla 1	Regla2	....	Regla N
Condiciones				
cond..1				
cond..2				
.....				
cond.. P				
Acciones				
Acción 1				
Acción 2				
.....				
Acción M				

Condiciones de  
entrada/Valores  
de entrada

Acciones que se  
deben tomar,  
dependiendo de  
las combinaciones  
de las condiciones

# Tablas de Decisión – Ejemplo 1

- Una compañía aseguradora de autos hace descuentos a los conductores que sean casados y/o buenos estudiantes. Los descuentos son los siguientes:
  - Sí es casado y buen estudiante, entonces el descuento es del 60%.
  - Sí es buen estudiante y no está casado, entonces el descuento es del 25%.
  - Sí es casado y no es buen estudiante, entonces el descuento es del 50%.
  - Sí no es casado ni buen estudiante, entonces no tiene descuento.

# Tablas de Decisión – Ejemplo 1

- Una compañía aseguradora de autos hace descuentos a los conductores que sean casados y/o buenos estudiantes. Los descuentos son los siguientes:
  - Sí es casado y buen estudiante, entonces el descuento es del 60%.
  - Sí es buen estudiante y no está casado, entonces el descuento es del 25%.
  - Sí es casado y no es buen estudiante, entonces el descuento es del 50%.
  - Sí no es casado ni buen estudiante, entonces no tiene descuento.
- **Se identifican las condiciones:**
  - Casado.
  - Buen estudiante.
- **Se identifica la acción/Resultados esperados:**
  - Descuentos.

# Tablas de Decisión

	Prueba 1	Prueba 2	Prueba 3	Prueba 4
Condición				
Casado??	Si	No	Si	No
Buen estudiante???	Si	Si	No	No
Acciones				
Descuento ( \$ )	60	25	50	0

# Tablas de Decisión

ID. Caso de Prueba	Casado?	Estudiante?	Resultado Esperado
CP1	Si	Si	Desc. 60
CP2	No	Si	Desc. 25
CP3	Si	No	Desc. 50
CP4	No	No	0

# Tablas de Decisión – Ejemplo 2

## Descripción de caso

Se quiere determinar el importe a facturar a los clientes de unos grandes almacenes según estos criterios:

- ✓ Si pagan con tarjeta oro tendrán un 15% de descuento.
- ✓ Si pagan con tarjeta club tendrán un 5% de descuento.
- ✓ Si la tarjeta (oro o club) es modalidad joven, tendrán un 5% de descuento.
- ✓ Los descuentos son acumulables.

### Condiciones:

- Tipo de Tarjeta?
- Modalidad?

### Acciones

- Descuento
- Calcular costo

# Tablas de Decisión: Ejemplo 2

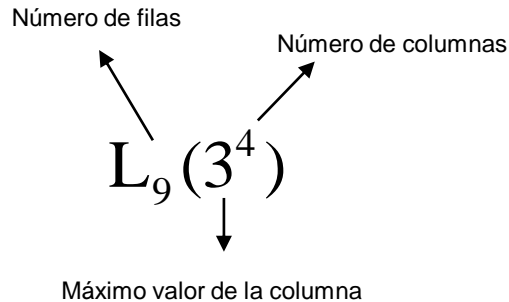
	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5
<b>Condición</b>					
Tipo de Tarjeta	Oro	Oro	Club	Club	Ninguna
Modalidad Tarjeta	Joven	<> Joven	Joven	<> Joven	Ninguna
<b>Acciones</b>					
Descuento ( %)	20%	15%	10%	5%	
Calcular importe	x	x	x	x	x



# Arreglos Ortogonales

## ¿Qué son arreglos Ortogonales?

Un arreglo ortogonal es un arreglo en 2 dimensiones, con la siguiente propiedad: Si se seleccionan 2 columnas cualquiera en el arreglo, todas las combinaciones de pares aparecerán en esas columnas.



# Arreglos Ortogonales

$L_9(3^4)$

	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Cuántos casos de prueba se tienen de acuerdo a la cantidad de variables y valores?

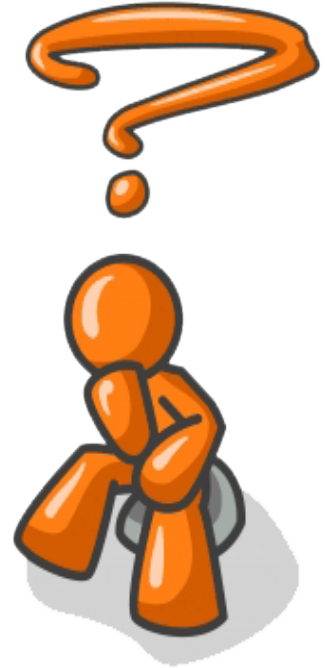
Posibles combinaciones  $(3 \times 3 \times 3 \times 3) = 81$ .

Los arreglos ortogonales solo garantizan que todas las combinaciones de pares existen en el arreglo

# Arreglos Ortogonales

## ¿Cuándo usar la técnica?

- En situaciones donde se tienen muchos casos de prueba para diseñar y ejecutar.
- No se tienen los suficientes recursos.
- La técnica ayuda a seleccionar un “buen subconjunto” de casos de prueba.



# Arreglos Ortogonales

## Situación 1.

Un sitio Web debe operar correctamente con diferentes browsers , usando diferentes plug – ins, corriendo sobre diferentes sistemas operativos en el cliente, utilizando diferentes servidores de aplicaciones y diferentes sistemas operativos en el servidor.

**Browsers:** IE 9, 10 y 11 Firefox52, 51, 50. Chrome 57 Opera 43.

**Plug-ins:** RealPlayer, MediaPlayer, ninguno.

**Sistemas operativos cliente:** Windows 10, 8, 7, XP, 2000, ME

**Servidor de Aplicaciones:** IIS, Apache y WebLogic

**Sistemas operativos servidor:** Windows NT, 2000 y Linux.

**Combinaciones:**  $8 \times 3 \times 6 \times 3 \times 3 = 1296$

# Arreglos Ortogonales

## Situación 2.

Un banco tiene un nuevo sistema de información listo para probar. El banco tiene diferentes clases de clientes, diferentes clases de cuentas y opera en diferentes estados del país, cada uno con diferentes regulaciones.

**Tipos de cliente:** Preferencial, afiliado, no afiliado, independiente

**Tipos de cuenta:** Ahorro, crédito, corriente, nómina, mixta, empresarial

**Estados:.** activa, inactiva, en mora, suspendida, en jurídico, en cambio

**Combinaciones:**  $4 \times 5 \times 6 = 120$

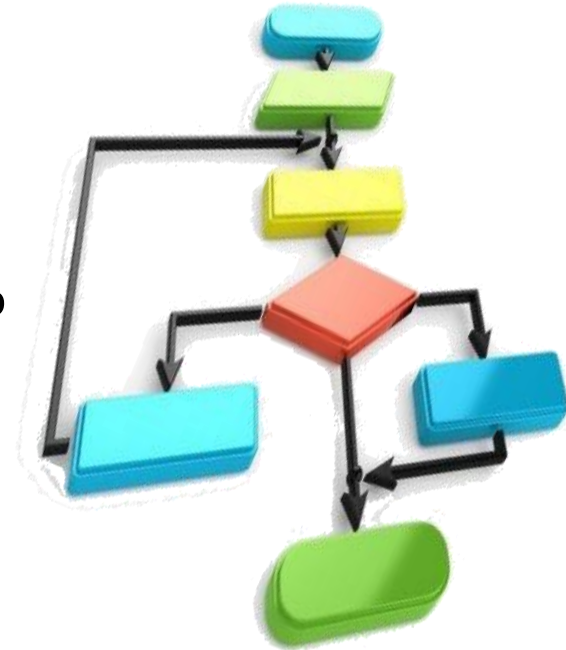
# Arreglos Ortogonales

## **Proceso para usar la técnica manualmente.**

- Identificar las variables/condiciones.
- Determinar el número de opciones de cada variable.
- Encontrar un arreglo ortogonal el cual tenga el tamaño requerido.

Ej:  $L_9(3^4)$

- Mapear el problema dentro del arreglo ortogonal.
- Construir los casos de prueba.



# Ejercicio

Un sitio Web debe operar correctamente en Chrome, safari y firefox, para Celulares, tablets y portátiles, con Windows, IOS y Android y además para las siguientes resoluciones: 1024x768, 768x480, 806x525.



# Ejercicio

1. Identificar las variables.

- Browsers
- Dispositivos
- Sistemas operativos
- Resoluciones

2. Determinar el número de opciones de cada variable.

- Browsers : 3
- Dispositivos: 3
- Sistemas Operativos: 3
- Resoluciones: 3

**Posibles escenarios de prueba (combinaciones):  $3 \times 3 \times 3 \times 3 = \underline{81}$**



# Ejercicio

3. Encontrar un arreglo ortogonal el cual tenga el tamaño requerido.

- Columnas (Variables) : 4.
- Máximo valor por columna (número de valores que puede tomar cada variable): 3 (1,2 ó 3).

$$3^4 \longrightarrow L_9(3^4)$$

	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

# Ejercicio

Mapear el problema dentro del Arreglo Ortogonal.

## Browser

1

1. Chrome.
2. Safari.
3. Firefox

## Dispositivos

2

1. Celulares
2. Tablets
3. Portátil

## Sistemas Operativos

3

1. Windows
2. IOS
3. Android

## Resolucion

4

1. 1024x768
2. 768x480
3. 806x525

	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

# Ejercicio

Mapear el problema dentro del Arreglo Ortogonal.

## Browser

1

1. Chrome.
2. Safari.
3. Firefox

## Dispositivos

2

1. Celulares
2. Tablets
3. Portátil

## Sistemas Operativos

3

1. Windows
2. IOS
3. Android

## Resolucion

4

1. 1024x768
2. 768x480
3. 806x525

	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

# Ejercicio

Mapear el problema dentro del Arreglo Ortogonal.

## Browser

1. Chrome.
2. Safari.
3. Firefox

## Dispositivos

1. Celulares
2. Tablets
3. Portátil

## Sistemas Operativos

1. Windows
2. IOS
3. Android

## Resolucion

1. 1024x768
2. 768x480
3. 806x525

	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

# Ejercicio

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>1</b>	1	1	1	1
<b>2</b>	1	2	2	2
<b>3</b>	1	3	3	3
<b>4</b>	2	1	2	3
<b>5</b>	2	2	3	1
<b>6</b>	2	3	1	2
<b>7</b>	3	1	3	2
<b>8</b>	3	2	1	3
<b>9</b>	3	3	2	1

# Ejercicio

4. Mapear el problema dentro del arreglo ortogonal.

	Browser	Dispositivo	S.O.	Resolución
1	Chrome	Celular	Windows	1024x768
2	Chrome	Tablet	IOS	768x480
3	Chrome	Portatil	Android	806x525
4	Safari	Celular	IOS	806x525
5	Safari	Tablet	Android	1024x768
6	Safari	Portatil	Windows	768x480
7	Firefox	Celular	Android	768x480
8	Firefox	Tablet	Windows	806x525
9	Firefox	MPortatil	IOS	1024x768

5. Construir los casos de prueba.

# Arreglos Ortogonales

## **Notas:**

- La técnica de arreglos ortogonales reduce significativamente el número de pruebas que deben diseñarse y ejecutarse.
- Pueden existir combinaciones dadas por el arreglo ortogonal que no son válidas.

# Arreglos Ortogonales

## *Proceso para usar la técnica utilizando una herramienta.*

- Identificar las variables/condiciones.
- Identificar valores para cada condición utilizando Técnicas de diseño de pruebas: PE y AV.
- Utilizar Allpairs para aplicar la técnica de Arreglos Ortogonales.
- **los valores negativos de una variable no se llevan al arreglo ortogonal, ya que generan combinaciones no válidas.**
- Una vez identificado los escenarios de prueba se realiza la especificación.





# Allpairs

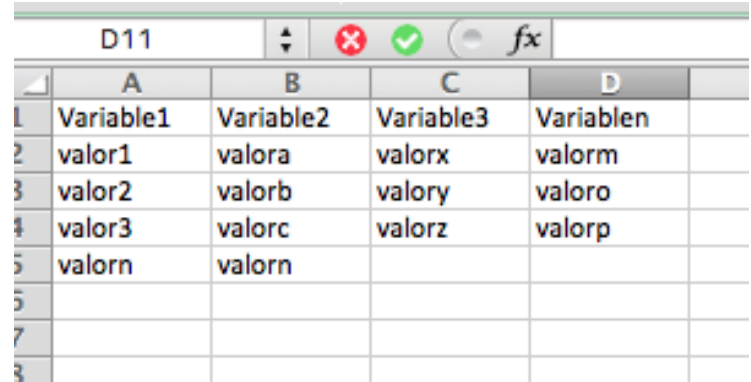
All Pairs es una herramienta que permite realizar una combinación efectiva de variables utilizando la técnica de Arreglos Ortogonales.

**Ofrece:**

- ✓ Combinaciones basadas en tablas estadísticas (AO)
- ✓ Combinación de variables sin importar la cantidad de valores que pueda tener cada una.
- ✓ Visualización de posibles escenarios.

# Uso de Allpairs

- ✓ Cuando identifiques las variables en el excel, asegúrate de no tener tildes ni caracteres especiales que puedan afectar el archivo.
- ✓ Inicia siempre en la Columna A1, con el nombre de la variable y debajo de la misma identifica los valores.
- ✓ Has lo mismo en cada columna con cada variable y valores.

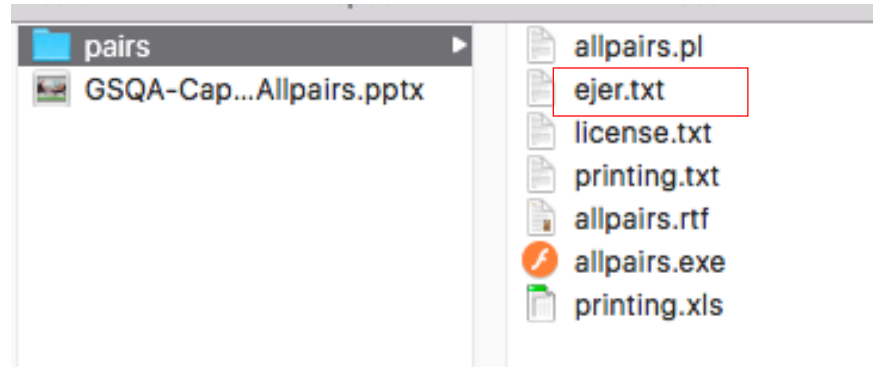


The screenshot shows an Excel spreadsheet with a table of data. The table has 5 columns labeled A, B, C, and D. The first row (A1) contains the variable names: Variable1, Variable2, Variable3, and Variablen. The subsequent rows (A2-A5) contain values for each variable: valor1, valor2, valor3, and valorn. The values for Variable2 are valora, valorb, valorc, and valorn. The values for Variable3 are valorx, valory, valorz, and valorp. The values for Variablen are valorm, valoro, and valorp. The table is displayed in a window titled 'D11' with standard Excel window controls.

	A	B	C	D
1	Variable1	Variable2	Variable3	Variablen
2	valor1	valora	valorx	valorm
3	valor2	valorb	valory	valoro
4	valor3	valorc	valorz	valorp
5	valorn	valorn		
6				
7				
8				

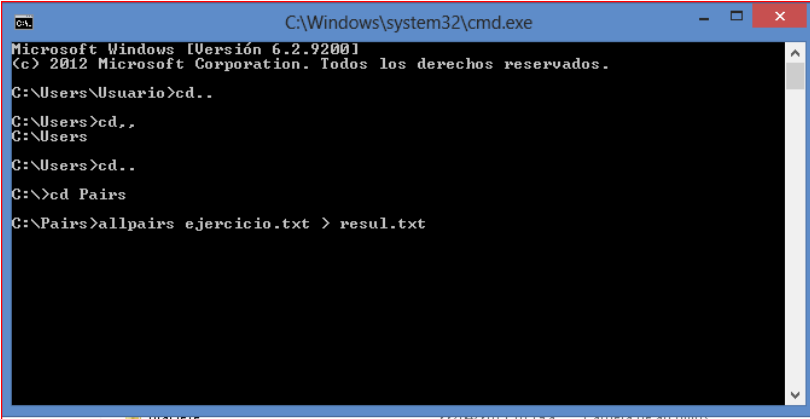
# Instrucciones

- Copia la carpeta PAIRS en la Raiz C de tu equipo.
- Una vez termines de identificar las variables y valores del requerimiento, guarda el archivo como: txt(MS-DOS) dentro de la carpeta PAIRS



## Uso de AllPairs

1. Abre una consola (CMD)
2. Ubícate en la carpeta C:\PAIRS



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.2.9200]
(c) 2012 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>cd..
C:\Users>cd..
C:\Users>cd..
C:\Users>cd..
C:\>cd Pairs
C:\Pairs>allpairs ejercicio.txt > resul.txt
```

3. Ejecuta el siguiente comando:

**ALLPAIRS** nombreakchivo.txt > resul.txt

- **Nombre archivo:** nombre del archivo que se guardo como txt (MS-DOS)
- **Resul:** nombre que se dará al archivo donde se generarán los resultados (puedes asignarle el que quieras)

# Uso de AllPairs

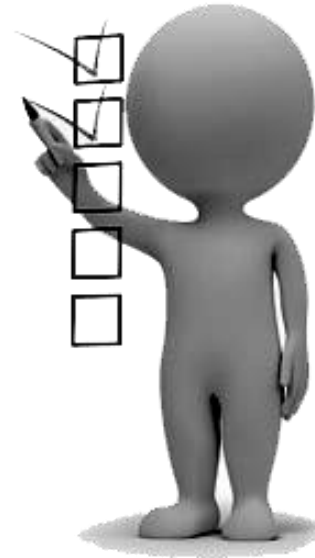
4. En la Carpeta PAIRS, se ha creado un archivo tipo texto, con el nombre que le asignaste al resultado de la combinación. Ej. Resul.txt
5. Abre el archivo:

```
TEST CASES
case  Factura Soportes      Torre  Materiales Existen  Carga  Flag  pairings
1      Si      Si      TS-TS  Si      Manual Activo 15
2      No      No      TS-TS  No      Automatica Inactivo 15
3      Si      No      TS-TN  Si      Automatica Activo 11
4      No      Si      TS-TN  No      Manual Inactivo 11
5      Si      Si      TN -TS No      Automatica Activo 8
6      No      No      TN -TS Si      Manual Inactivo 8
7      Si      ~No    ~TS-TS ~No    ~Manual Inactivo 1
8      No      ~Si    ~TS-TS ~Si    ~Automatica Activo 1
```

- El archivo muestra una serie de combinaciones de acuerdo a las Variables y Valores especificados, con el titulo TEST CASES.
- Estas combinaciones se convierte en casos de pruebas en la Matriz de casos de prueba. A cada uno se le debe agregar su respectivos pasos y resultados esperados.

# Ten en cuenta que...

1. Cuando uno de los valores presente este símbolo: ~ , significa que ese valor puede ser cambiado por cualquier otro valor de la variable.
2. En ningún caso se puede eliminar una combinación.
3. De ser necesario puedes adicionar alguna combinación específica siempre y cuando sea de interés específico para el cliente.
4. Las variables y valores identificadas y con las cuales se generó el Arreglo ortogonal, debe quedar almacenado en la pestaña de análisis de las MCP.



# Ejercicio

- Realice los ejercicios propuestos en el taller de Técnica de arreglos ortogonales
- Tiempo estimado 40 minutos
- Solución en grupo al finalizar el tiempo estimado.

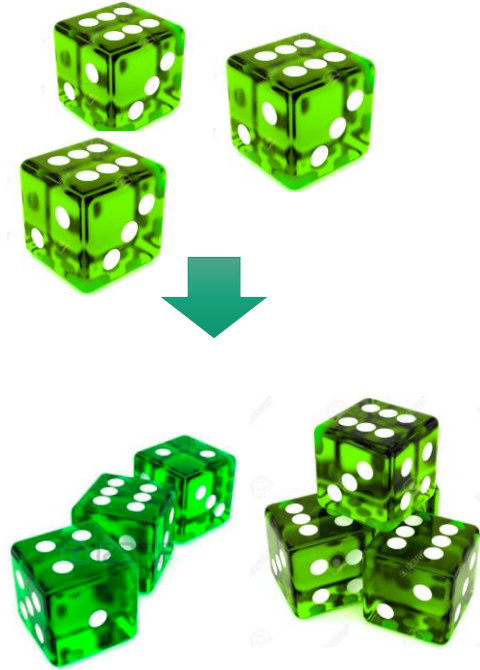


## 2. Definición de Escenarios de Prueba (Resumen)



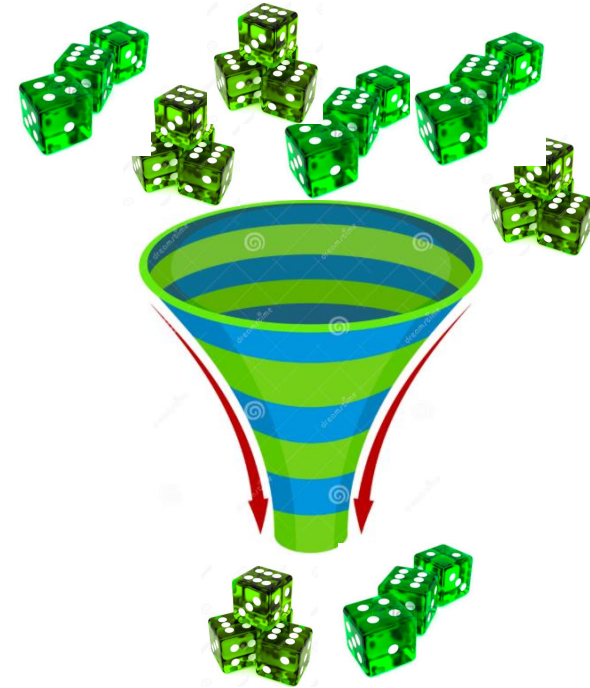
# Definición de escenarios de pruebas

- Por cada funcionalidad que hace parte del alcance, **identificar las variables relevantes**.
- Utilizar las técnicas de diseño de pruebas de primer nivel (Partición de Equivalencia – Análisis de Valor Limite) para la **identificación de valores que afectan el resultado de cada una de las funcionalidades**.
- Aplicación de técnicas del segundo nivel (Tablas de Decisión y Arreglos Ortogonales) para la **identificación de la muestra de escenarios de prueba**.
- **Identificación de resultados esperados genéricos y específicos** por cada una de las funcionalidades.



# Definición de escenarios de pruebas

- **Opcional:** Realizar el calculo de ahorro en el proceso de diseño y ejecución de pruebas.
- **Importante:** Validar el diseño de pruebas con los usuarios líderes involucrados en cada funcionalidad.



# Ejemplo Uso de Técnicas

# Ejemplo 1

- **Funcionalidad a probar:**
- **Retiro de una cuantía de dinero de cuentas de ahorros**, considerando la reglamentación legal de la Superintendencia Financiera, que impone el Cobro de un impuesto a retiros mayores a 60 UVT (Unidad de Valor Tributario ~ 5% del SMLV)
- **Técnica de diseño de pruebas a aplicar:**
- Tablas de Arreglos Ortogonales

# Ejemplo 1

## Condiciones de Prueba

Identificación y clasificación de las variables de entrada al proceso.

The diagram illustrates the classification of input variables. A tree structure originates from the top center, branching down to the first three columns of the table: 'Modalidad', 'Ubicación Retiro', and 'Tipo de cuenta'. These three columns are grouped under the label 'Partición de Equivalencia'. A separate line points from the label 'Análisis Valor Limite' to the fourth column, 'Valor Transaccion'.

Modalidad	Ubicación Retiro	Tipo de cuenta	Valor Transaccion
Efectivo	Local	Individual	menor 60 UVT
Cheque	nacional	Alternativa	60 UVT
Nota Debito	Intenacional	Conjunta	mayor 60 UVT

## Una prueba Exhaustiva implicaría

Posibles combinaciones  $(3 \times 3 \times 3 \times 3) = 81$  condiciones de Prueba.

# Ejemplo

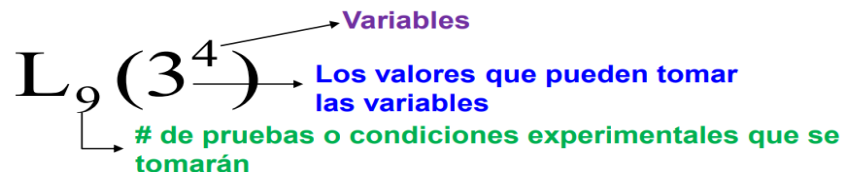
## Técnica Recomendada: Tablas de Arreglos Ortogonales

### ¿Por qué arreglos?

- La cantidad de variables que se ven comprometidas en las pruebas y la cantidad de valores que pueden tomar cada uno de ellos.

### Proceso

- Identificar las variables.
  - Determinar el número de opciones de cada variable.
  - Encontrar un arreglo ortogonal el cual tenga el tamaño requerido (ya existen !!!)
- Mapear el problema dentro del arreglo ortogonal.
- Construir los requerimientos de prueba.



# Ejemplo

**Identificar las variables** ( ... son 4 en este caso)

- Modalidad
- Ubicación de Retiro
- Tipo Cuenta
- Valor Transacción

**Combinaciones:  $3 \times 3 \times 3 \times 3 = 81$**

**1.a Determinar el número de opciones de cada variable.**

- Modalidad :3
- Ubicación de Retiro: 3
- Tipo Cuenta: 3
- Valor Transacción: 3

# Ejemplo 1


## 1.b Encontrar un arreglo ortogonal el cual tenga el tamaño requerido.

- Columnas (Variables) : 4.
- Máximo valor por columna

(número de valores que puede tomar cada variable): **3** (1,2 ó 3)

$$3^4 \longrightarrow L_9(3^4)$$

Variables



	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1



# Ejemplo 1

- 2. Mapear el problema dentro del Arreglo Ortogonal.

**Modalidad** 1

1- Efectivo  
2- Cheque  
3- Nota Debito

**Valor Transacción** 4

1- Menor 60 UVT  
2- 60 UVT  
3- Mayor 60 UVT

**Ubicación de Retiro** 2

1- Local  
2- Nacional  
3- Internacional

**Tipo de Cuenta** 3

1- Individual  
2- Alterna  
3- Conjunta

	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

# Ejemplo 1

## 2. Mapear el problema dentro del Arreglo Ortogonal.

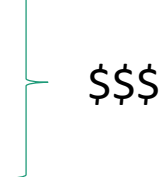
	Modalidad	Ubicación Retiro	Tipo de cuenta	Valor Transaccion
1	Efectivo	Local	Individual	Menor 60 UVT
2	Efectivo	Nacional	Alternativa	60 UVT
3	Efectivo	Internacional	Conjunta	Mayor 60 UVT
4	Cheque	Local	Alternativa	Mayor 60 UVT
5	Cheque	Nacional	Conjunta	Menor 60 UVT
6	Cheque	Internacional	Individual	60 UVT
7	Nota Debito	Local	Conjunta	60 UVT
8	Nota Debito	Nacional	Individual	Mayor 60 UVT
9	Nota Debito	Internacional	Alternativa	Menor 60 UVT

## 3. Construir casos de prueba:

Realizar un retiro en efectivo local con una cuenta de ahorros Individual por un valor menor a los 60 UVT.

# Ejemplo

¿Cuál es el ahorro con este diseño de pruebas?

- Esfuerzo en Diseño
  - Esfuerzo en Planeación
  - Esfuerzo en Ejecución de Pruebas
- 

¿Cuánto es el ahorro en este caso?

- De 81 posibles casos (100%),
- se selecciona una muestra de 9 casos, (11%)
- Ahorro equivalente al 89%

# Ejemplo 2: Aplicación de Técnicas de Diseño

## Funcionalidad: Realizar Pago de Factura

### 1. Identificación de las variables y valores de entrada al proceso

*Partición de Equivalencia*

**Una prueba Exhaustiva implicaría**

Posibles combinaciones ( $3 \times 6 \times 2$ ) = 36 escenarios de Prueba.

Tipo Cliente	Forma de Pago	Transacción
VIP	Efectivo	Pago por documento
Empresarial	Débito automático	Pago por suscripción.
Life Line	Cheque	
	Tarjeta débito	
	Tarjeta crédito	
	Sin definir.	

# Aplicación de Técnicas: Tablas y Arreglos Ortogonales

**Porque arreglos?** la cantidad de variables que se ven comprometidas en las pruebas y la cantidad de valores que pueden tomar cada uno de ellos.

- Identificar las variables.
- Determinar el número de opciones de cada variable.
- Encontrar un arreglo ortogonal el cual tenga el tamaño requerido (ya existen !!!)
- Ej:
- Mapear el problema dentro del arreglo ortogonal.
- Construir los Requerimientos de prueba.

$$L_{18}(3^1 6^1 2^1)$$

Variables

Los valores que pueden tomar las variables

# Aplicación de Técnicas de Diseño

- **Identificar las variables** ( ... son 3 en este caso)
  - Tipo cliente
  - Forma de pago
  - Transacción
- **Determinar el número de opciones de cada variable.**
  - Tipo cliente : 3
  - Forma de pago : 6
  - Transacción : 2
- Combinaciones:  $3 \times 6 \times 2 = 36$


# Aplicación de Técnicas de Diseño

**Encontrar un arreglo ortogonal el cual tenga el tamaño requerido.**

- Columnas (Variables) : 3.
- Máximo valor por columna (número de valores que puede tomar cada variable).
- Arreglo Mixto – cada variable con diferente número opciones.

$$(3^1 6^1 2^1) \longrightarrow L_{18}(3^1 6^1 2^1)$$

Variables



	1	2	3
1	1	1	1
2	1	2	2
3	1	3	1
4	1	4	2
5	1	5	1
6	1	6	2
7	2	1	1
8	2	2	2
9	2	3	1
10	2	4	2
11	2	5	1
12	2	6	2
13	3	1	1
14	3	2	2
15	3	3	1
16	3	4	2
17	3	5	1
18	3	6	2

# Aplicación de Técnicas de Diseño

**Mapear el problema dentro del Arreglo Ortogonal.**

## Tipo Cliente

1

- 1 - VIP.
- 2 - Empresarial.
- 3 - Life Line

## Forma de Pago

2

- 1 - Efectivo.
- 2 - Débito automático.
- 3 - Cheque.
- 4 - Tarjeta débito.
- 5 - Tarjeta crédito.
- 6 - Sin definir.

## Transacción

3

- 1 - Pago por documento.
- 2 - Pago por suscripción.

	1	2	3
1	1	1	1
2	1	2	2
3	1	3	1
4	1	4	2
5	1	5	1
6	1	6	2
7	2	1	1
8	2	2	2
9	2	3	1
10	2	4	2
11	2	5	1
12	2	6	2
13	3	1	1
14	3	2	2
15	3	3	1
16	3	4	2
17	3	5	1
18	3	6	2



# Aplicación de Técnicas de Diseño

Mapear el problema dentro del arreglo ortogonal.

	Tipo cliente	Forma de pago	Transacción
1	VIP	Efectivo	Pago por documento
2	Empresarial	Efectivo	Pago por suscripción
3	VIP	Débito automático	Pago por suscripción
4	Empresarial	Débito automático	Pago por documento
5	Life Line	Cheque	Pago por documento
6	VIP	Cheque	Pago por suscripción
7	Life Line	Tarjeta débito	Pago por suscripción
8	VIP	Tarjeta débito	Pago por documento
9	Empresarial	Tarjeta crédito	Pago por documento
10	Life Line	Tarjeta crédito	Pago por suscripción
11	Empresarial	Sin definir	Pago por suscripción
12	Life Line	Sin definir	Pago por documento
13	Life Line	Efectivo	Pago por documento
14	Life Line	Débito automático	Pago por suscripción
15	Empresarial	Cheque	Pago por documento
16	Empresarial	Tarjeta débito	Pago por suscripción
17	VIP	Tarjeta crédito	Pago por documento
18	VIP	Sin definir	Pago por suscripción

# Aplicación de Técnicas de Diseño

## 5. Construir casos de pruebas

- Se construyen los 18 requerimientos de pruebas a partir de las combinaciones de
- valores de variables generados por el arreglo ortogonal.
- **Casos de Prueba**
- *Para un plan de telefonía realizar un pago de factura con cheque, utilizando un documento o cupón de pago, para un cliente Empresarial.*

# Resultados Esperado por Requerimiento de Prueba

- Para mapear los resultados esperados con las condiciones de prueba, utilizamos la técnica de tablas de decisión.
- Son utilizadas para registrar reglas del negocio complejas basadas en un conjunto de condiciones, las cuales deben ser implementadas en el sistema.
- Se deben identificar:
  - Reglas. (Condiciones de prueba)
  - Acciones (Resultados esperados **Genericos y Especificos**).
- ❖ Resultados Genericos: Son aquellos resultados que independiente del valor que tome una variable de la funcionalidad, este siempre se presenta.
- ❖ Resultados Especificos: Son aquellos resultados que pertenecen a un valor de variable.

# Aplicación de Técnicas de Diseño

## **Acciones Genericas:**

**AG 1-** Verificar la afectación de la Cartera del cliente

**AG 2-** Verificar registro del pago – conciliación

## **Acciones Especificas - Producto**

**AE4** -Verificar que el cuadre de cajas tenga en cuenta el valor pagado

## **Acciones Especificas – Cliente Empresarial**

**AE5** -Verificar la distribucion del pago en los productos asociados al cliente empresarial.

## **Acciones Especificas - Cupon**

**AE1** – Verificar que se actualice el reporte de control cupones de pago.

**AE2** - Verificar que el cupón de pago sea tenido en cuenta en la conciliacion bancaria

**AE3** - Verificar que el pago efectuado corresponde al valor registrado en el cupón

# Aplicación de Técnicas de Diseño de Pruebas

Con las condiciones de prueba y acciones identificadas se procede a construir la tabla de decisión:

	Reglas	
	15	6
<b>Condiciones</b>		
<b>Tipo cliente</b>		
VIP	No	Sí
Empresarial	Sí	No
Life Line	No	No
<b>Forma de pago</b>		
Efectivo	No	No
Débito automático	No	No
Cheque	Sí	Sí
Tarjeta débito	No	No
Tarjeta crédito	No	No
Sin definir	No	No
<b>Transacción</b>		
Pago por documento	Sí	No
Pago por suscripción	No	Sí
<b>Resultado esperado</b>		
AG 1- Verificar la afectación de la Cartera del cliente	Sí	Sí
AG 2- Verificar registro del pago - conciliacion	Sí	Sí
AE1 – Verificar que se actualice el reporte de control cupones de pago.	Sí	Sí
AE2 - Verificar que el cupon de pago sea tenido en cuenta en la conciliacion bancaria	Sí	No
AE3 - Verificar que el pago efectuado corresponde al valor registrado en el cupon	Sí	No
AE4 -Verificar que el cuadro de cajas tenga en cuenta el valor pagado	No	Sí
AE5 -Verificar la distribucion del pago en los productos asociados al cliente empresarial.	Sí	No

# Aplicación de Técnicas de Diseño de Pruebas

**Realizar un pago de factura con cheque, utilizando un documento o cupon de pago, para un cliente Empresarial.**


## Resultados:

- **AG 1-** Verificar la afectación de la Cartera del cliente
- **AG 2-** Verificar registro del pago – conciliación
- **AE1** – Verificar que se actualice el reporte de control cupones de pago.
- **AE2** - Verificar que el cupón de pago sea tenido en cuenta en la conciliación bancaria
- **AE3** - Verificar que el pago efectuado corresponde al valor registrado en el cupón
- **AE5** -Verificar la distribución del pago en los productos asociados al cliente empresarial.

	Reglas	
	15	6
<b>Condiciones</b>		
<b>Tipo cliente</b>		
VIP	No	Si
Empresarial	Si	No
Life Line	No	No
<b>Forma de pago</b>		
Efectivo	No	No
Débito automático	No	No
Cheque	Si	Si
Tarjeta débito	No	No
Tarjeta crédito	No	No
Sin definir	No	No
<b>Transacción</b>		
Pago por documento	Si	No
Pago por suscripción	No	Si
<b>Resultado esperado</b>		
AG 1- Verificar la afectación de la Cartera del cliente	Si	Si
AG 2- Verificar registro del pago - conciliación	Si	Si
AE1 – Verificar que se actualice el reporte de control cupones de pago.	Si	Si
AE2 - Verificar que el cupon de pago sea tenido en cuenta en la conciliación bancaria	Si	No
AE3 - Verificar que el pago efectuado corresponde al valor registrado en el cupon	Si	No
AE4 -Verificar que el cuadro de cajas tenga en cuenta el valor pagado	No	Si
AE5 -Verificar la distribución del pago en los productos asociados al cliente empresarial.	Si	No

# Aplicación de Técnicas de Diseño de Pruebas

Cual es el ahorro con este diseño de pruebas?

- Esfuerzo en Diseño
  - Esfuerzo en Planeación
  - Esfuerzo en Ejecución de Pruebas
- 

Cuanto es el ahorro en este caso?

- De 36 posibles casos (100%),
- se selecciona una muestra de 18 casos, (50%)
- Ahorro equivalente al 50%

# 3. Especificación de Casos de Prueba



# Estructura de un caso de prueba

- Descripción de cada uno de los casos de prueba, detallando como mínimo:

IDENTIFICADOR	DESCRIPCIÓN DEL CASO DE PRUEBA	PRECONDICIONES	PASO A PASO	RESULTADOS ESPERADOS
Identifica el caso de prueba, un caso de prueba se diferencia de otro caso a través de este identificador.	Debe especificar la funcionalidad y la combinación de variable y valor de variable que conforma el escenario	Descripción de los datos, parámetros o procesos previos que deben considerarse antes de la ejecución del caso de prueba.	Descripción del paso a paso que se debe realizar para ejecutar el caso de prueba	Describir el listado de los resultados esperados de la ejecución de la funcionalidad.



The diagram consists of five 3D rectangular blocks arranged in an ascending staircase pattern from left to right. Each block is labeled with a number in the top right corner: 01 (green), 02 (blue), 03 (green), 04 (blue), and 05 (orange). These numbers correspond to the five columns of the table above.



# Especificación del caso de prueba

Titulo	Descripción del caso y Pasos de prueba	Resultados Esperados
Titulo del caso indicando la funcionalidad a evaluar y su principal condición.	Funcionalidad a evaluar y sus condiciones.	
	Pre-requisitos	
	Paso 1	Resultado esperado paso 1
	Paso 2	Resultado esperado paso 2
	Paso 3	Resultado esperado paso 3
Titulo del caso indicando la funcionalidad a evaluar y su principal condición.	Funcionalidad a evaluar y sus condiciones.	
	Pre-requisitos	
	Paso 1	Resultado esperado paso 1
		Resultado esperado paso 1
	Paso 2	Resultado esperado paso 2
		Resultado esperado paso 2
	Paso 3	Resultado esperado paso 3
Titulo del caso indicando la funcionalidad a evaluar y su principal condición.	Funcionalidad a evaluar y sus condiciones.	
	Pre-requisitos	
	Paso 1	
	Paso 2	Resultado esperado paso 2
		Resultado esperado paso 2
	Paso 3	Resultado esperado paso 3

Titulo	Descripción del caso y Pasos de prueba	Resultados Esperados
<b>2. Cargue de aviso de recibo -</b> <b>Compañía: 1 - Archivo:</b> <b>Encabezado incorrecto</b>	<p>Cargue de aviso de recibo cuando:</p> <p>Compañía: 1  Archivo: Encabezado incorrecto  Número de remisión (AAJ): Invalido (para el cliente y la ORDER)  Número de orden de compra (ON): Valido (para el cliente y la remisión)  Cliente (Seg NAD_DP - EAN): Valido (orden de compra y remisión)  Documento con FLAG_REMISION_RECIBO en: S  Estado del documento: 3R (DOC.STATUS = 3 y DOC.FLAGAUT = R)  Artículo (LIN - Identificación del ítem) : existente en el documento  Cantidad aviso de recibo (QTY 194): Igual a DET_VENTAS.CANT_REMISIONADA  Correo configurado: Sí</p> <p>Precondiciones:</p> <ul style="list-style-type: none"> <li>* Tener los datos de acceso del aplicativo Delta y acceso a la base de datos.</li> <li>* Tener acceso a las carpetas (IN y OUT) dispuestas para la recepción del aviso de recibo.</li> <li>* Contar con los avisos de recibo en formato .HSE.</li> <li>* Los archivos deben contener mínimo 3 artículos: incluir mínimo 3 segmentos LIN.</li> <li>* Haber ejecutado el cargue del archivo de aviso de recibo.</li> <li>* Tener como parámetro el EAN de la compañía (tabla PARAMETROS).</li> <li>* Configurar tres (3) correos (tabla de contactos para email) de la siguiente forma: <ol style="list-style-type: none"> <li>1. con bodega</li> <li>2. con bodega y cia</li> <li>3. sin parámetros (ni bodega ni cia)</li> </ol> </li> </ul>	
	Ingresar con los datos de acceso al aplicativo Delta.	El usuario ingresa al sistema
	Ingresar a la ruta Ventas/Transacciones EDI/Manejo de aviso de recibo/Consultas/Consultar de aviso de recibo.	
	Consultar el aviso de recibo.	En la pestana encabezado, verificar que el estado (ESTADO_COMPARA) se encuentre en 3 - Aviso totalmente inconsistente.
		Para los artículos, en la pestana Detalle, verificar que el estado (ESTADO_DET_COMPARA ) se encuentre en 1 - Documento no existe.
	Ingresar a base de datos y consultar las tablas AVIREC_ENC y AVIREC_DETALLE.	Verificar que el campo AVIREC_ENC.ESTADO_COMPARA se encuentre en 3.
		Para cada uno de los artículos enviados, verificar que el estado compara se encuentre en 1.
		Verificar que el sistema envíe el email al correo configurado.
		Verificar que el asunto del email cumpla con la siguiente estructura:
		- AVISO DE RECIBO: ##### ORDEN DE COMPRA ##### REMISION: ##### Cliente: ##### - Discrepancias
		Verificar que el cuerpo del email cumpla con la siguiente estructura:
		- Señor JJJJJJ, del cliente XXXXXX se recibió el Aviso de Recibo ##### de la Orden de compra y Remisión ##### del documento: CIA: SUC: BOD: CPTO: FUENTE: con discrepancias, el estado del AVISO es ## (ESTADO_ENC_COMPARA).

# Ejercicio

- Aplicar técnicas de diseño de prueba y realizar Matriz de casos de prueba.
- Socialización del ejercicio.



