

# Pruebas del Software: Pruebas Funcionales

**Análisis & Diseño de Sistemas**  
**M.G.P. Álvaro Mena Monge**



UNIVERSIDAD DE COSTA RICA  
SEDE DEL ATLÁNTICO

# Agenda

- Las pruebas funcionales
  - Pruebas de Unidad
  - Pruebas de Sistema
- Enfoque de Pruebas
  - Casos de Prueba
  - Procedimientos de Prueba
  - Scripts de Prueba
  - Escenarios de Prueba

# Las pruebas funcionales

- Pruebas basadas en una perspectiva de comportamiento o de caja negra
- Desatiende cualquier conocimiento de cómo opera internamente la unidad/sistema
- Etapas de las pruebas funcionales
  - Pruebas de Unidad
  - Pruebas de Sistema

- Se refiere a realizar pruebas sobre *unidades de programa en aislamiento*
- Las unidades puede ser vistas como
  - Funciones, métodos o procedimientos
  - En la P.O.O. es la clase
- Naik y Priyadarshi [2008] definen las unidades de programa como:
  - “una pieza de código, tal como una función o método de clase, que es invocada desde afuera de la unidad y que puede invocar otras unidades de programa”

# Pruebas de Unidad

- Como las unidades de programa implementan funciones, es natural realizar pruebas antes de integrarla con otras unidades
  - Razón por la cual se debe probar en *aislamiento*
- Estas pruebas son aplicadas por el desarrollador que escribió el programa

# Pruebas de Unidad

- Razones para probar en aislamiento
  - Los errores detectados se atribuyen a esa unidad y por ende fácilmente modificados
  - Es necesario verificar cada ruta de ejecución en la unidad → implica una selección cuidadosa de los datos de entrada

# Pruebas de Unidad

- Las unidades deben ser probadas como sigue:
  - Ejecutar cada línea de código
  - Ejecutar cada predicado en la unidad para evaluarlos como verdadero o falso separadamente

## Pruebas de Unidad – "Estáticas"

- “El software debe someterse a una fase de inspección o corrección en cada milestone”
- Técnicas de revisión aplicadas \* Revisar Naik y Priyadarshi [2008]
  - **Inspección**: Revisión en parejas de un producto de trabajo con una lista de cotejo predefinida
  - **Ir-a-través-de** (walkthrough): el autor guía al equipo a través de una ejecución manual o simulada del producto utilizando varios escenarios

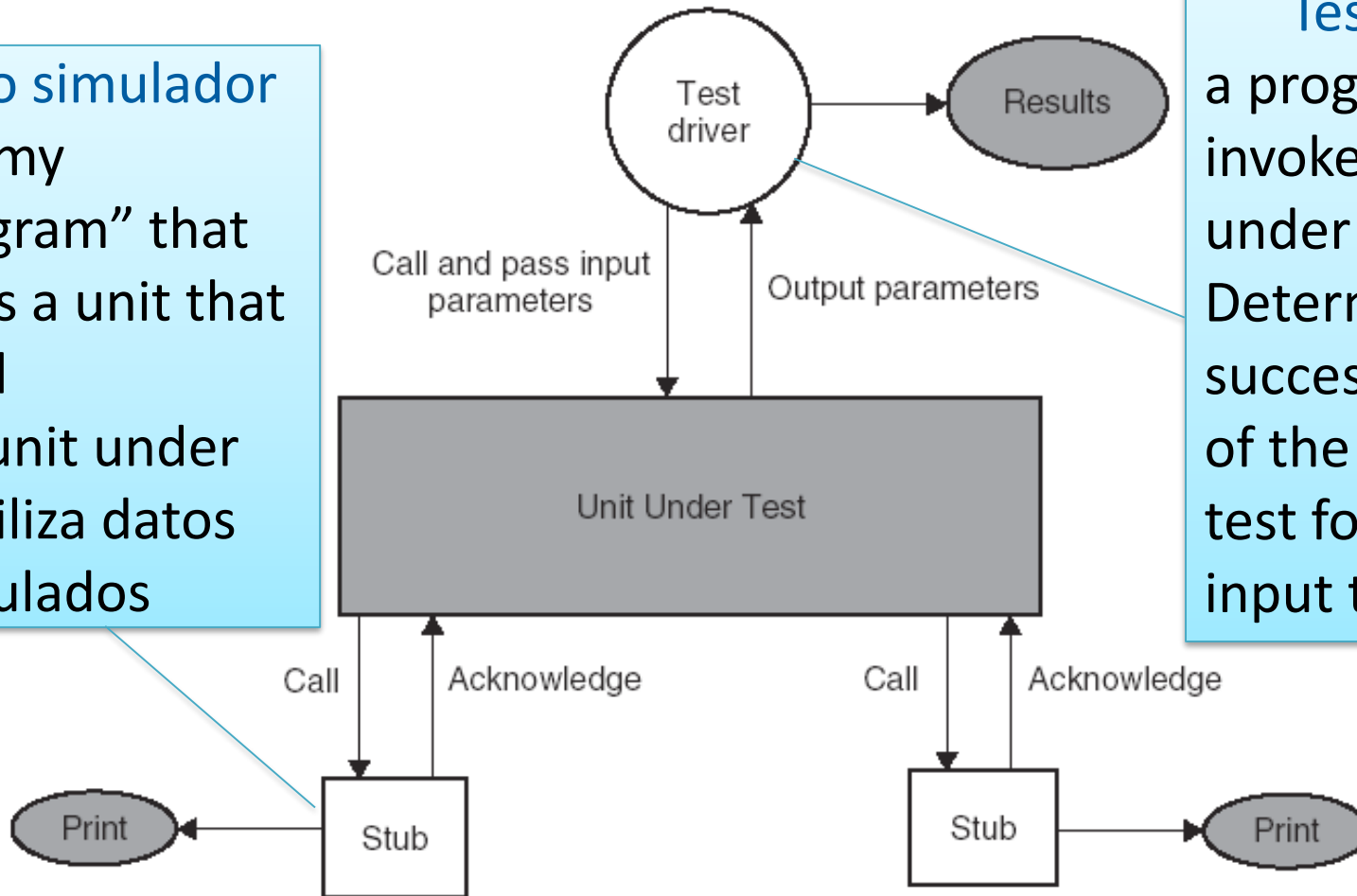


# Pruebas de Unidad - Dinámicas

- Pruebas de unidad basadas en ejecución
- Particularidades
  - La unidad es extraída del ambiente de ejecución actual
  - El ambiente de ejecución actual es emulado escribiendo más código (ambos deben compilar juntos)
  - El agregado de compilación es ejecutado con entradas seleccionadas
- Los resultados son recolectados mediante
  - Archivos de bitácoras, observación de pantalla y software de instrumentación especializado

# Pruebas de Unidad – Dinámicas

**Stubs o simulador**  
a “dummy subprogram” that replaces a unit that is called by the unit under test. Utiliza datos precalculados



**Test Driver**  
a program that invokes the unit under test. Determine the success or failure of the unit under test for each input test data.

***Ambiente de P. U. Dinámicas***

- Considera la solución total como una unidad
- Las pruebas del sistema se ejecutan para los casos de uso más importantes en una iteración específica como para los casos de uso anteriores (Pruebas de Regresión)



## Caso de Prueba

- Manassis [2003] define un caso de prueba como:
  - “un conjunto de entradas de datos, condiciones de ejecución y resultados esperados”
- RUP [2003]
  - “conjunto de entradas de prueba, condiciones de ejecución, y resultados esperados, identificados con el objetivo de evaluar algunos aspectos particulares de un ítem de prueba destino”



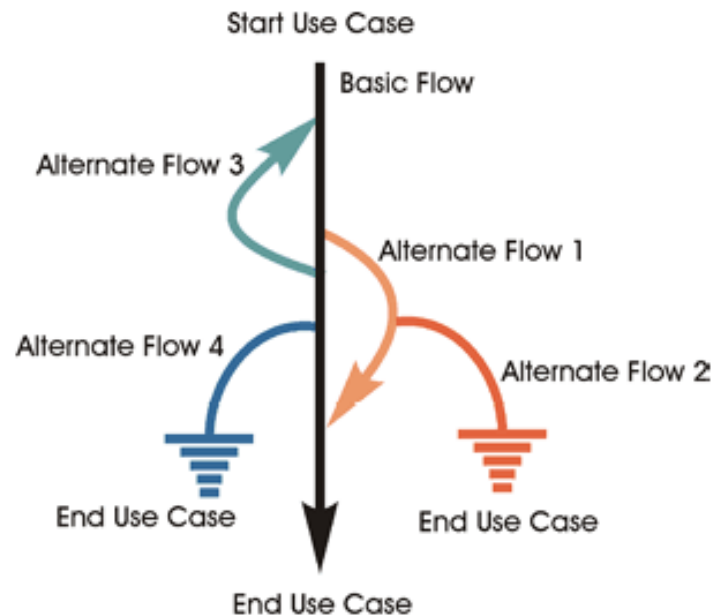
## Caso de Prueba

- Los C.P. proceden de *casos de uso*, *características de rendimiento* y *los riesgos* que afectan al proyecto
- En el caso de los casos de uso
  - definidos una vez que los casos de uso hallan sido especificados
  - representan escenarios o rutas completas a través de un caso de uso.



## Caso de Prueba: Escenarios de C.U.

- **Escenario:** instancia de un caso de uso o “ruta” completa a través de un caso de uso
  - Los usuarios finales pueden ir a través de muchas rutas mientras ejecutan la funcionalidad especificada en el caso de uso
  - Ej. Seguir el flujo básico podría ser un escenario





# Caso de Prueba: Escenarios de C.U.

## Register For Courses

### *Basic Flow*

#### 1. Logon

This use case starts when a Student accesses the Wylie University Web site. The system asks for, and the Student enters, the student ID and password.

#### 2. Select 'Create a Schedule'

The system displays the functions available to the student. The student selects "Create a Schedule."

#### 3. Obtain Course Information

The system retrieves a list of available course offerings from the Course Catalog System and displays the list to the Student.

#### 4. Select Courses

The Student selects four primary course offerings and two alternate course offerings from the list of available course offerings.

### *Alternate Flows*

#### 1. Unidentified Student

**In Step 1** of the Basic Flow, Logon, if the system determines that the student ID and/or password is not valid, an error message is displayed.

#### 2. Quit

The Course Registration System allows the student to quit at any time during the use case. The Student may choose to save a partial schedule before quitting. All courses that are not marked as "enrolled in" are marked as "selected" in the schedule. The schedule is saved in the system. The use case ends.

#### 3. Unfulfilled Prerequisites, Course Full, or Schedule Conflicts

In Step 5 of the Basic Flow, Submit Schedule, if the system determines that prerequisites for a selected course are not satisfied, that the course is full, or that there are schedule conflicts, the system will not enroll the student in the course. A message is displayed that the student can select a different course. The use case continues at Step 4, Select Courses, in the basic flow.



# Caso de Prueba: Escenarios de C.U.

## Register For Courses

### *Basic Flow*

#### 1. Logon

This use case starts when a Student accesses the Wylie University Web site. The system asks for, and the Student enters, the student ID and password.

#### 2. Select 'Create a Schedule'

The system displays the functions available to the student. The student selects "Create a Schedule."

#### 3. Obtain Course Information

The system retrieves a list of available course offerings from the Course Catalog System and displays the list to the Student.

#### 4. Select Courses

The Student selects four primary course offerings and two alternate course offerings from the list of available course offerings.

### *Alternate Flows*

#### 4. Course Catalog System Unavailable

In Step 3 of the Basic Flow, Obtain Course Information, if the system is down, a message is displayed and the use case ends.

#### 5. Course Registration Closed

If, when the use case starts, it is determined that registration has been closed, a message is displayed, and the use case ends.





## Caso de Prueba: Escenarios de C.U.

- Lista de escenarios posibles del caso de uso “Register for Courses”

Scenario 1	Basic Flow			
Scenario 2	Basic Flow	Alternate Flow 1		
Scenario 3	Basic Flow	Alternate Flow 1	Alternate Flow 2	
Scenario 4	Basic Flow	Alternate Flow 3		
Scenario 5	Basic Flow	Alternate Flow 3	Alternate Flow 1	
Scenario 6	Basic Flow	Alternate Flow 3	Alternate Flow 1	Alternate Flow 2
Scenario 7	Basic Flow	Alternate Flow 4		
Scenario 8	Basic Flow	Alternate Flow 3	Alternate Flow 4	

# Generación de Casos de Prueba

## Generación de Casos de Prueba

- Leer el C.U. e identificar los escenarios (combinación de flujo básico y flujos alternos)
- Crear la Matriz de Escenarios

Determinar los escenarios del C.U.

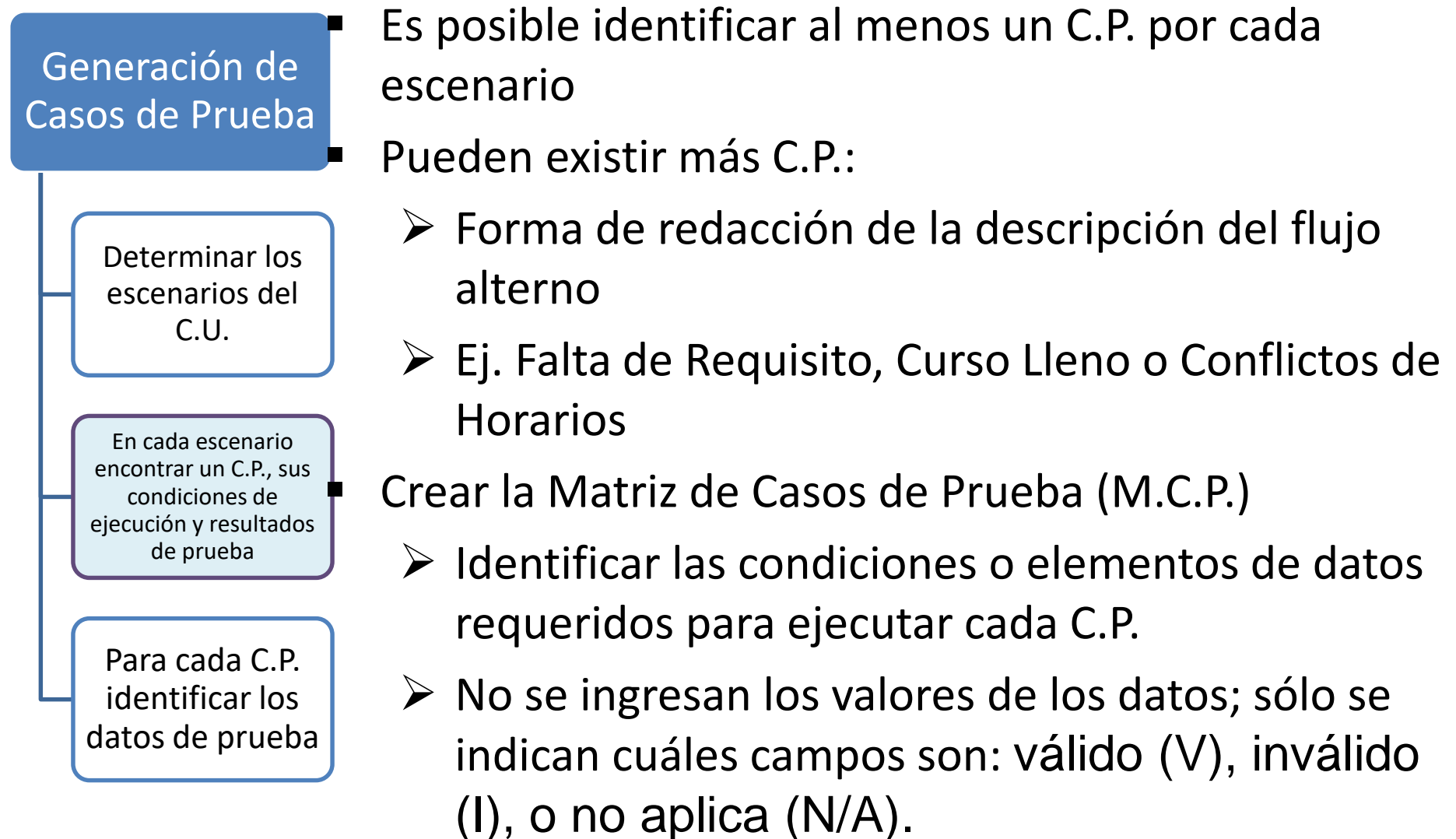
En cada escenario encontrar un C.P., sus condiciones de ejecución y resultados de prueba

Para cada C.P. identificar los datos de prueba

Scenario Name	Starting Flow	Alternate
Scenario 1 - Successful registration	Basic Flow	
Scenario 2 - Unidentified student	Basic Flow	A1
Scenario 3 - User quits	Basic Flow	A2
Scenario 4 - Course catalog system unavailable	Basic Flow	A4
Scenario 5 - Registration closed	Basic Flow	A5
Scenario 6 – Cannot enroll	Basic Flow	A3

*Matriz de escenarios parcial para el C.U. "Register for Courses"*

# Generación de Casos de Prueba



# Generación de Casos de Prueba

Test Case ID	Scenario/Condition	Student ID	Password	Courses selected	Pre -requisites fulfilled	Course Open	Schedule Open	Expected Result
RC 1	Scenario 1- Successful registration	V	V	V	V	V	V	Schedule and confirmation number displayed
RC 2	Scenario 2- Unidentified student	I	N/A	N/A	N/A	N/A	N/A	Error message; back to login screen
RC 3	Scenario 3- Valid user quits	V	V	N/A	N/A	N/A	N/A	Login screen appears
RC 4	Scenario 4- Course registration system unavailable	V	V	N/A	N/A	N/A	N/A	Error message; back to step 2
RC 5	Scenario 5- Registration closed	V	V	N/A	N/A	N/A	N/A	Error message; back to step 2
RC 6	Scenario 6- Cannot enroll -- course full	V	V	V	V	I	V	Error message; back to step 3
RC 7	Scenario 6- Cannot enroll -- prerequisite not fulfilled	V	V	V	I	V	V	Error message; back to step 4
RC 8	Scenario 6- cannot enroll -- schedule conflict	V	V	V	V	V	I	Error message; back to step 4

Matriz de Casos de Prueba para el C..U. "Register for Courses"

# Generación de Casos de Prueba

## Generación de Casos de Prueba

Determinar los escenarios del C.U.

En cada escenario encontrar un C.P., sus condiciones de ejecución y resultados de prueba

Para cada C.P. identificar los datos de prueba

Definir los datos de prueba en lugar de las indicaciones de las condiciones de prueba.

La misma plantilla de la matriz es usada para este propósito

# Generación de Casos de Prueba

Test Case ID	Scenario/Condition	Student ID	Password	Courses selected	Pre -requisites fulfilled	Course Open	Schedule Open	Expected Result
RC 1	Scenario 1- Successful registration	jheumann	abc123	M101> E201 S101	Yes	Yes	Yes	Schedule and confirmation number displayed
RC 2	Scenario 2- Unidentified student	Jheuman1	N/A	N/A	N/A	N/A	N/A	Error message; back to login screen
RC 3	Scenario 3- Valid user quits	jheumann	abc123	N/A	N/A	N/A	N/A	Login screen appears
RC 4	Scenario 4- Course registration system unavailable	jheumann	abc123	N/A	N/A	N/A	N/A	Error message; back to step 2
RC 5	Scenario 5- Registration closed	jheumann	abc123	N/A	N/A	N/A	N/A	Error message; back to step 2
RC 6	Scenario 6- Cannot enroll -- course full	jheumann	abc123	M101> E201 S101	Yes	M101 Full	Yes	Error message; back to step 3
RC 7	Scenario 6- Cannot enroll -- prerequisite not fulfilled	jheumann	abc123	M101> E201 S101	Not for E201	Yes	Yes	Error message; back to step 4
RC 8	Scenario 6- cannot enroll -- schedule conflict	jheumann	abc123	M101> E201 S101	yes	Yes	E201 and S101 conflict	Error message; back to step 4

Matriz de Casos de Prueba (Datos de entrada) para el C..U. "Register for Courses"

# Procedimientos de Prueba

- Definen los pasos y las condiciones del entorno (sistema y aplicación) para ejecutar casos de prueba
- Describen actividades seguidas por el verificador para configurar, ejecutar y analizar una prueba
  - Incluye: definición de los valores de los datos, archivos E/S, pruebas automatizadas a ejecutar.
- Escritos para pruebas manuales y/o automatizadas

# Procedimientos de Prueba

- Propósitos del artefacto:
  - Cómo configurar el ambiente de prueba
  - Dónde encontrar el conjunto de datos de prueba
  - Los pasos para ejecutar las pruebas
  - Qué hacer con los resultados de prueba



# Procedimientos de Prueba

Plantilla Procedimiento de Prueba (1)	
Nombre	Nombre para identificarlo
Casos de prueba relacionados	Identificación de los Casos de Prueba que cubre el procedimiento
Estado inicial del sistema antes que el caso de prueba inicie	Relacionado con la aplicación y con el entorno del sistema. Ej. Conexión a un lector de código de barra, habilitar un puerto HTTP.
Estado de los datos usados por el sistema antes que el caso de prueba empiece	Descripción de los pasos para preparar los datos y referenciar algún script automatizado u otro documento que describe el procedimiento para preparar los datos
Pasos del procedimiento de prueba	Requiere ser bien específico. Ej. Referenciar botones o enlaces a presionar, campos de entrada a llenar.

# Procedimientos de Prueba

## Plantilla Procedimiento de Prueba (2)

Estado final esperado del sistema después que el caso de prueba finalice

Complementario a la descripción de los resultados del caso de prueba.  
Es opcional; se puede enfocar en aspectos del entorno del sistema no considerados en el caso de uso

Estado esperado de los datos usados por el sistema luego de finalizado el caso de prueba

Puede involucrar referenciar la forma como extraer los datos del sistema.  
Ej. Una sentencia SQL a ejecutar

# Scripts de Prueba

- Elemento usado para probar la funcionalidad de un aplicación de software
- Scripts manuales:
  - El verificador deberá ejecutarlos
  - Deben estar incrustados en los procedimientos de prueba
- Scripts automatizados:
  - Programas de software escritos para probar el sistema
  - Referenciados en los procedimientos de prueba

# Escenarios de Prueba

- Representan usos plausibles del sistema por parte de los usuarios
  - Enfocado en las tareas que el usuario desea ejecutar
- Útiles para *describir sesiones de usuario típicas*
  - Involucra una secuencia de casos de uso
  - Ej. Si un usuario compra libros, el escenario está compuesto por: Autenticar Usuario, repetición n-aria de (Revisar el Catálogo, Seleccionar Libros), Terminar Compra.
- Los casos de uso son los bloques de construcción de los escenarios de prueba
  - El escenario está integrado por más de un caso de prueba

# Escenarios de Prueba

## Plantilla Escenario de Prueba (1)

<b>Nombre</b>	Nombre que permita identificarlo
<b>Propósito</b>	Describe el razonamiento por el cual el escenario fue definido
<b>Descripción</b>	Describe lo que el escenario hace
<b>Tipo de prueba</b>	Funcional/Load/Estrés/Performance/Seguridad ....
<b>Test cases used</b>	Lista de nombres de los casos de prueba involucrados
<b>Estado de los datos usados por el sistema antes que el caso de prueba empiece</b>	

# Escenarios de Prueba

## Plantilla Escenario de Prueba (2)

### Pasos del escenario de pruebas

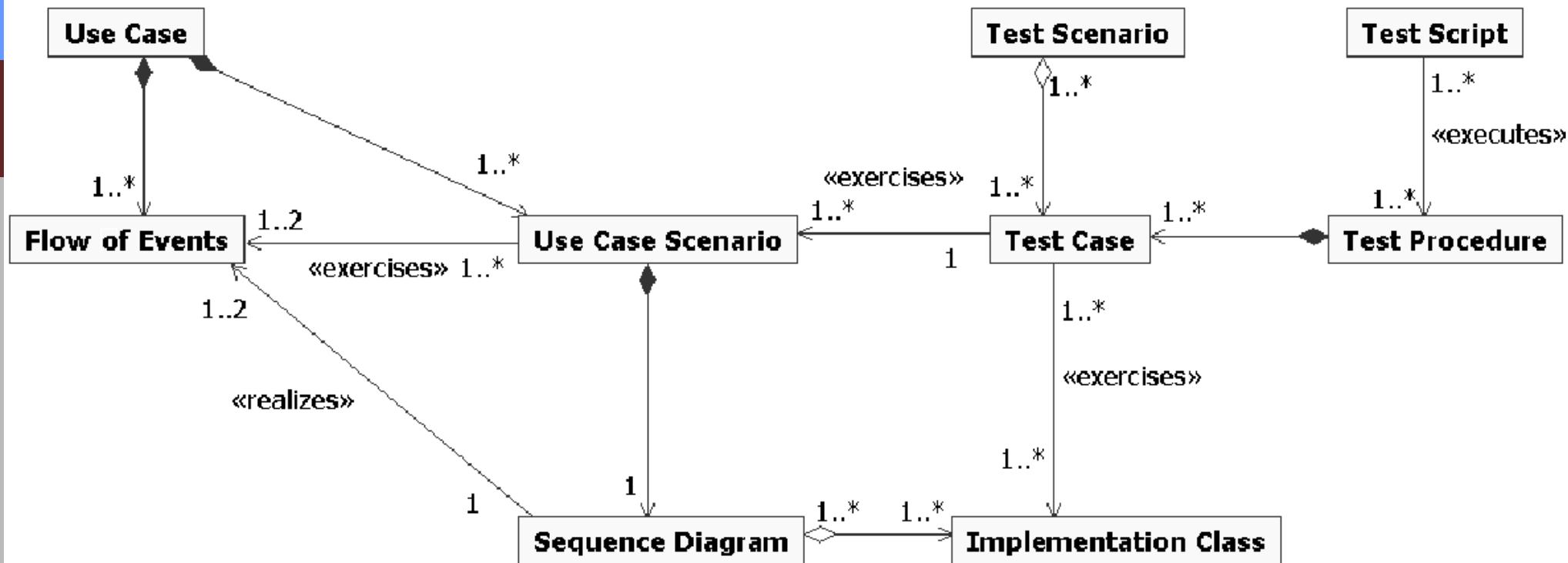
Enumere los casos de prueba que deben ser aplicados. Identifique el conjunto de repeticiones de casos de prueba y cuántas. Ej.

- 1.Ejecute Caso de Prueba A  
Ejecute 2 seguido de 3 (7 veces)
- 2.Ejecute Caso de Prueba B
- 3.Ejecute Caso de Prueba C
- 4.Ejecute Caso de Prueba D

### Directivas adicionales e información por recolectar

Cualquier detalle extra sobre la ejecución de pruebas

# Conceptos discutidos y sus relaciones



# Referencias

- [Manassis 2003] Manassis, Enricos. **Practical Software Engineering: Analysis and Design for the .NET Platform** Addison Wesley, 2003 – *Cap. 9 Testing*
- [Naik & Priyadarshi 2008] Naik, Kshirasagar; Tripathy, Priyadarshi. **Software testing and quality assurance: theory and practice**. John Wiley & Sons, USA, 2008 \*  
*Cap. 3 Unit Testing*  
*Página 90 – Lista de cotejo del código*
- Pressman, Roger S. **Ingeniería del software: un enfoque práctico**. Sexta Edición. McGraw-Hill, USA, 2005



## Ejercicio caso de prueba

- Elegir el principal caso de uso de su proyecto
- Crear matriz de escenarios
- Crear matriz de caso de prueba