



Ingeniería

Desarrollo de Software

Sesión #2

Aseguramiento de la Calidad

jessica.hernandez@umi.edu.mx



academiaglobal



TEMARIO

- Objetivo
- Resumen temas
- Herramientas
- Material de apoyo
- Pregunta de sesión

Objetivos:

Diseñar Casos de Prueba.

Diseñar Matrices de Prueba.

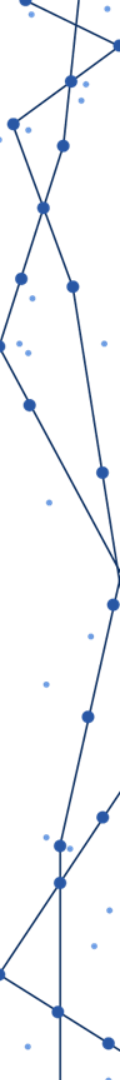
Hoy aprenderemos a ...

- **Diseñar Casos de Prueba.**
- **Diseñar Matrices de Prueba.**
- Prueba de caja blanca y sus tipos
- Estrategia de pruebas
- Pruebas de aplicaciones Orientadas o Objetos
- UML
- Casos de Uso
- Pruebas para aplicaciones Web



- La prueba de caja blanca:

Asegura la operación interna del programa, revisa las rutas lógicas a través del software y las colaboraciones entre componentes, es decir, se comprueban los caminos lógicos del programa, así como las condiciones y ciclos, examinando el estado del programa en varios puntos.

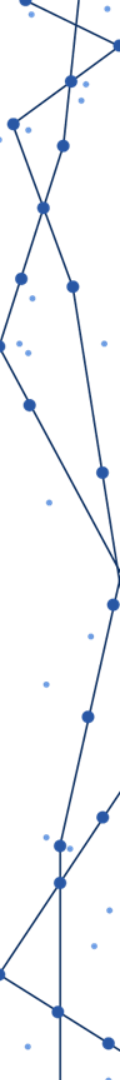


Existen 2 tipos de pruebas de caja blanca:

- De estructura de datos locales.
Estudia las variables del programa. Que esté declarada y no repetida, ni declarada local o globalmente. Que haya referencias a todas las variables.
- De cobertura lógica. (Se subdivide a su vez en 6)

- De cobertura lógica: Se subdivide a su vez en 6:
 - 1.- De cobertura de sentencias: Comprueba que todas las sentencias se ejecuten al menos una vez.
 - 2.- De cobertura de decisión: Comprueba que cada decisión se pruebe al menos una vez, "verdadero" (true) o "false" (false)
 - 3.- De cobertura de condición: Comprueba decisiones "verdadero" (true) o "false" (false).

- De cobertura lógica: Se subdivide a su vez en 6:
 - 4.- De cobertura de condición de decisión: Comprueba condición y decisión a la vez.
 - 5.- De cobertura de condición múltiple: Cada decisión multi condición se traduce en condición simple, aplicando posteriormente la cobertura de decisión.
 - 6.- De cobertura de caminos: Comprueba todos los caminos de un programa.



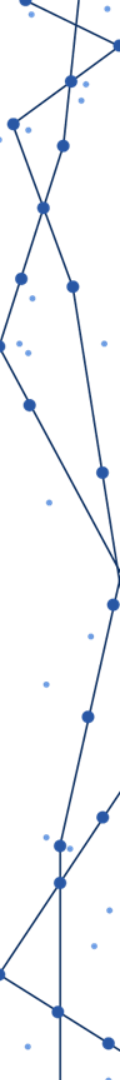
- Estrategia de pruebas

Deben seguirse de manera secuencial:

- Prueba de unidad
- Prueba de integración
- Prueba de validación
- Prueba de sistema



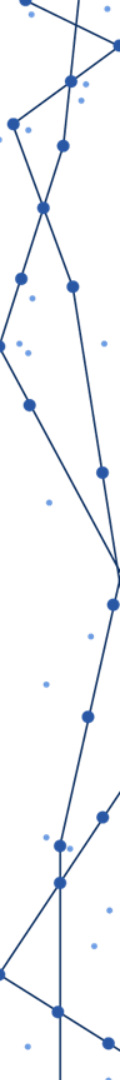
- Pruebas no convencionales
 - Pruebas de aplicaciones Orientadas o Objetos
 - Pruebas para aplicaciones Web



- Pruebas no convencionales
 - Pruebas de aplicaciones Orientadas o Objetos

Análisis Orientadas o Objetos:

- Modelo de Casos de Uso: Interacción entre usuario y sistema.
- Modelo de Dominio: Vocabulario implementado en el sistema.

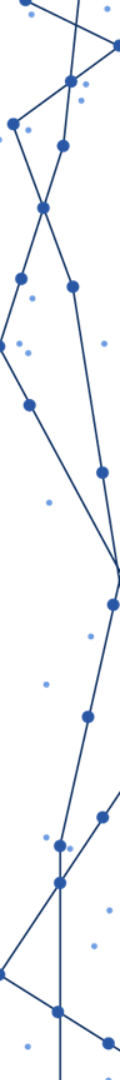


Lenguaje Unificado de Modelado “UML”

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

Se lo puede definir como un lenguaje gráfico para:

- Visualizar
- Especificar
- Construir
- Documentar un sistema

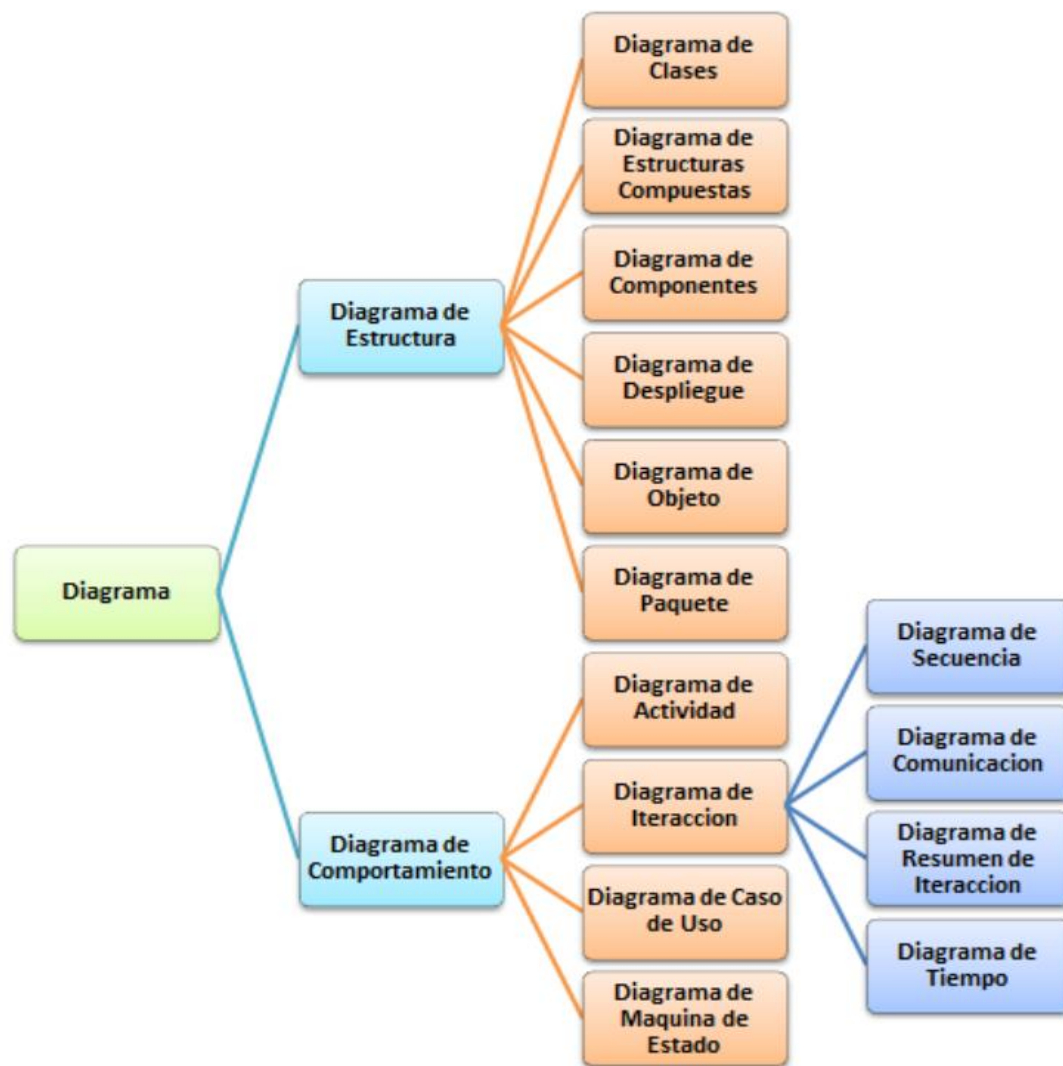


Lenguaje Unificado de Modelado “UML”

UML ofrece un estándar para describir un “plano” del sistema (modelo), incluyendo aspectos conceptuales tales como:

- Procesos de negocio
- Funciones del sistema
- Expresiones de lenguajes de programación
- Esquemas de bases de datos

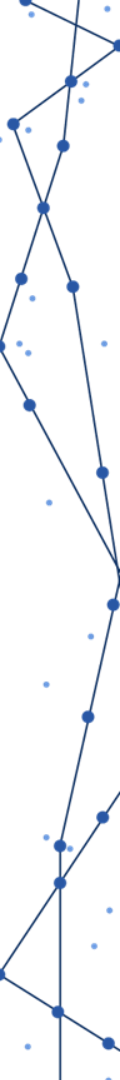




Diagramas de Casos de Uso

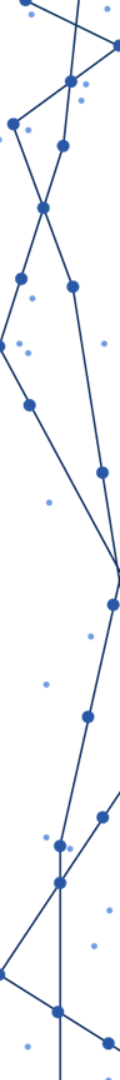
Los diagramas de caso de uso modelan la funcionalidad del sistema usando **actores** y **casos de uso**.

Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios.

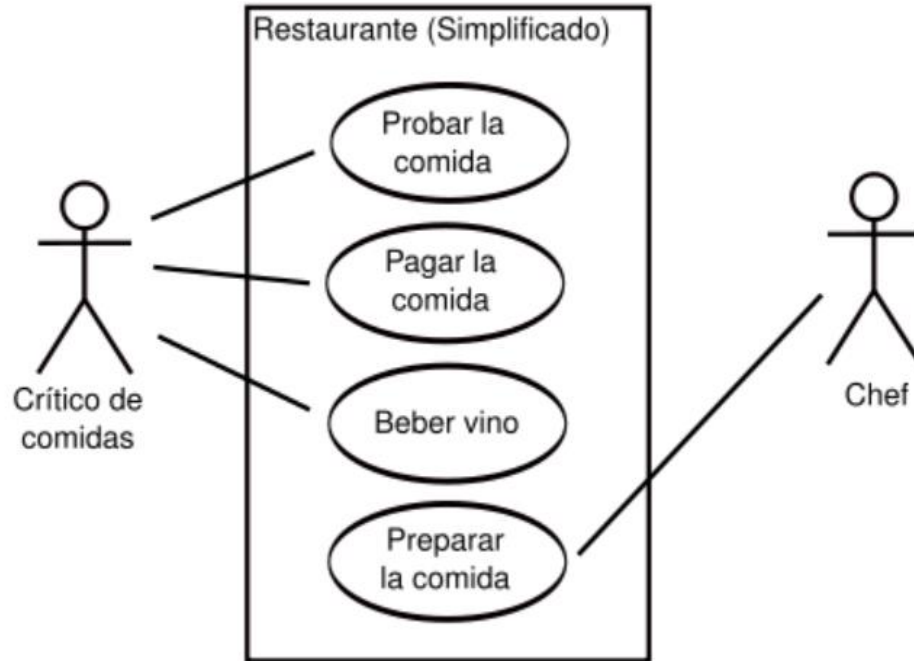


Símbolos de los casos de uso

Sistema: El rectángulo representa los límites del sistema que contiene los casos de uso. Los actores se ubican fuera de los límites del Sistema.



Símbolos de los casos de uso



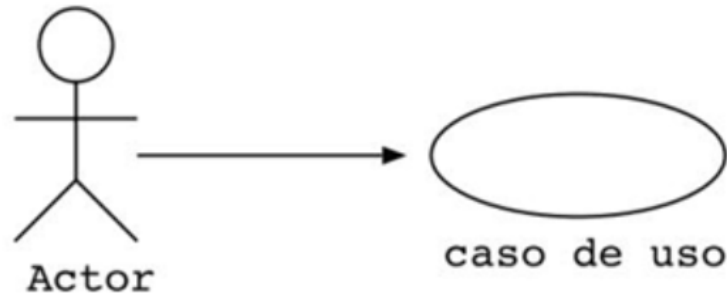
Símbolos de los casos de uso

Caso de uso: Se representan con óvalos. La etiqueta en el óvalo indica la función del sistema.



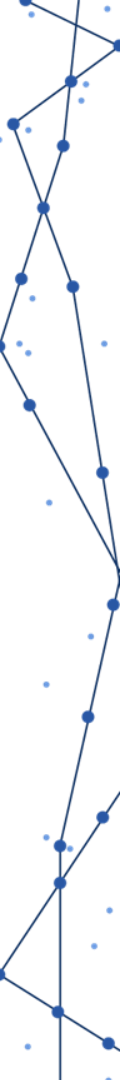
Símbolos de los casos de uso

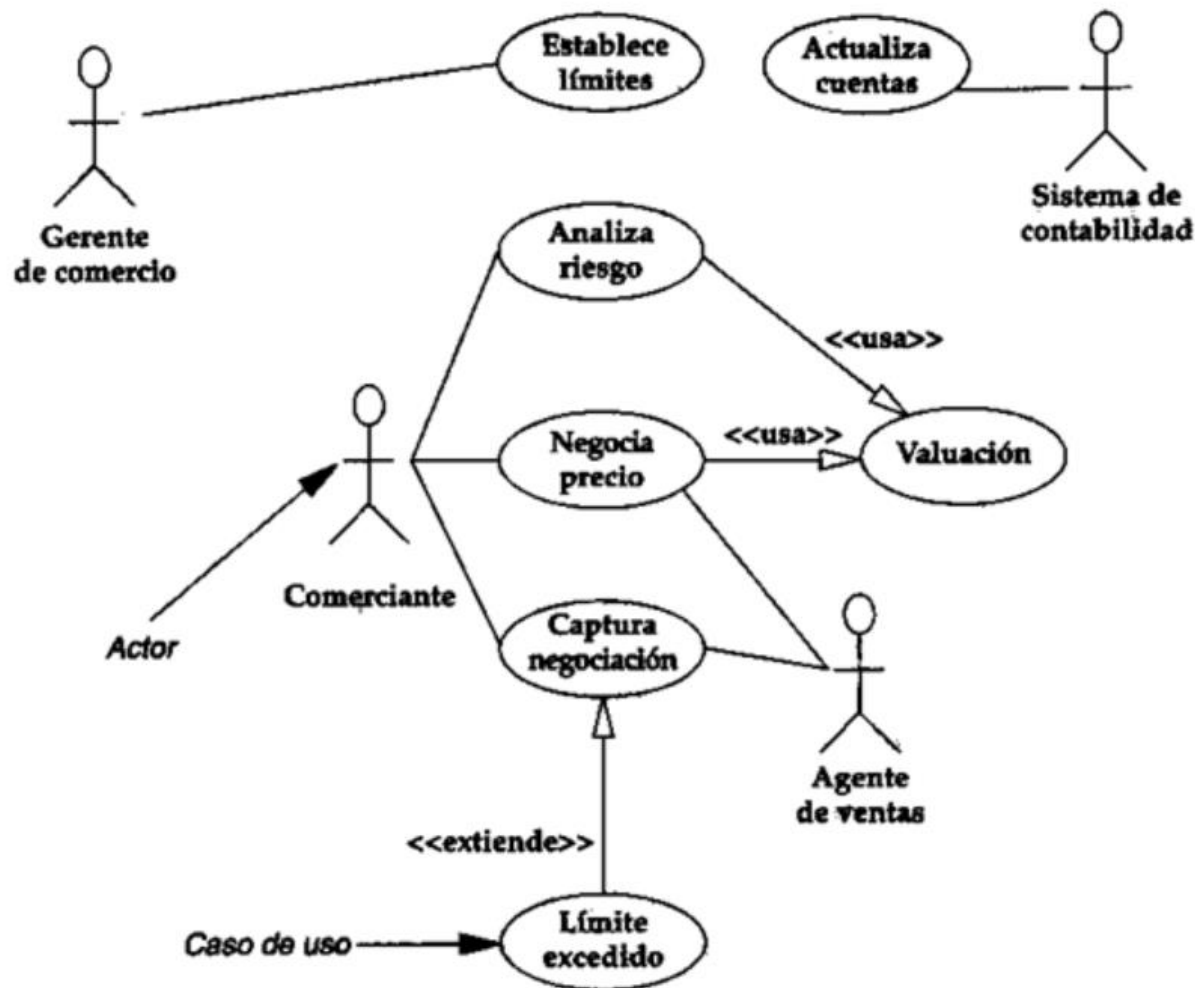
Actor: Un diagrama de caso de uso contiene los símbolos del actor y del caso de uso, junto con líneas conectoras. Los actores son similares a las entidades externas; existen fuera del sistema. El término actor se refiere a un rol específico de un usuario del sistema.



Símbolos de los casos de uso

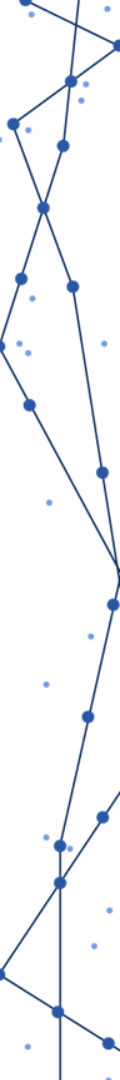
Relaciones: Las relaciones entre un actor y un caso de uso, se dibujan con una línea simple. Para relaciones entre casos de uso, se utilizan flechas etiquetadas “incluir” o “extender.” Una relación “incluir” indica que un caso de uso es necesitado por otro para poder cumplir una tarea. Una relación “extender” indica opciones alternativas para un cierto caso de uso.






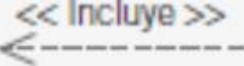
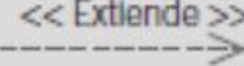

Símbolos de los casos de uso

Relaciones de los casos de uso: Las relaciones activas se conocen como relaciones de comportamiento y se utilizan principalmente en los diagramas de casos de uso. Hay cuatro tipos básicos de relaciones de comportamiento: comunica, incluye, extiende y generaliza.



Símbolos de los casos de uso

Relaciones de los casos de uso

Relación	Símbolo	Significado
Comunica		Para conectar un actor con un caso de uso se utiliza una línea sin puntas de flecha.
Incluye		Un caso de uso contiene un comportamiento común para más de un caso de uso. La flecha apunta al caso de uso común.
Extiende		Un caso de uso distinto maneja las excepciones del caso de uso básico. La flecha apunta del caso de uso extendido al básico.
Generaliza		Una "cosa" de UML es más general que otra "cosa". La flecha apunta a la "cosa" general.

Documentación de los casos de uso

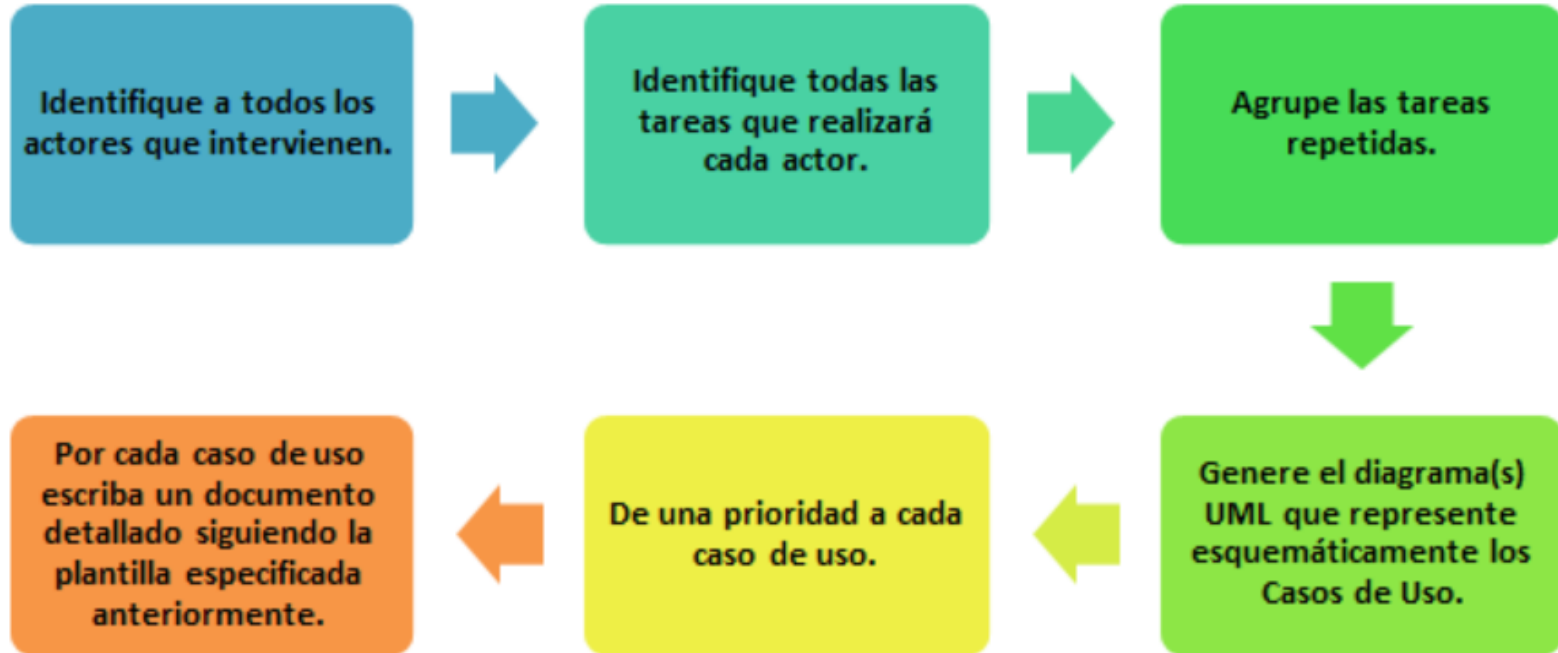
Existen dos formas principales de documentar un caso de uso:

- Un diagrama en UML
- Un documento detallado

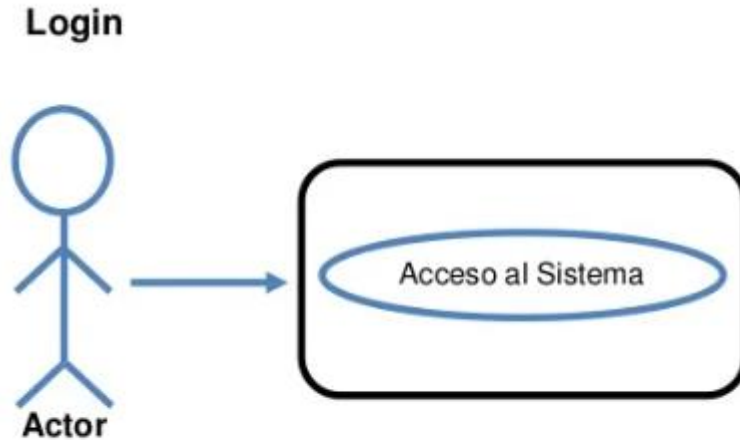
Documentar casos de usos no es una tarea fácil que se pueda dominar de un día para otro, requiere de tiempo, disciplina y experiencia, sin embargo podemos definir una serie de pasos identificables para escribir los casos de uso.



Pasos para la Documentación de los casos de uso:



Pasos para la Documentación de los casos de uso: Ejemplo:



Pasos para la Documentación de los casos de uso: Ejemplo:

DESCRIPCION DEL PROCESO
<ul style="list-style-type: none">• Nombre: Acceso al sistema• Propósito. Ingresar al sistema entrando a un navegador de internet e ingresar a la pagina donde se ubica el modulo de acceso.• Precondición. Que se ingrese la dirección donde se encuentre el modulo de acceso.• Pos condición. El usuario podrá acceder al módulo de acceso e ingresar los datos que le pidan como son: nombre de usuario y password• Reglas del negocio. Solo los usuarios registrados podrán ingresar al sistema.• Resumen. El remitente ingresa a internet por medio de un navegador e introduce dirección donde se encuentra alojado el acceso al sistema.

Pasos para la Documentación de los casos de uso: Ejemplo:

Actores	Actividades del sistema
1.- El usuario ingresa al módulo modificaciones	
	2.-Muestra formulario
3.-El usuario llena formulario	
	4.-Valida existencia
	5.-Muestra formulario de datos
6.-Usuario modifica datos	
	7.-Genera modificación
	8,-Envia el mensaje de modificación exitosa
9-Recibe el mensaje de modificación exitosa	



Caso de Prueba, ejemplo:

Nombre del Proyecto: (Nombre del proyecto en el que se está trabajando)

CASO DE PRUEBA

Prioridad de prueba: (baja/media/alta): (Urgencia de la prueba)

Prueba diseñada por: (Nombre del probador)

Nombre del módulo: (Módulo del sistema que se va a revisar: Login, registro de usuarios, etc)

Fecha del diseño de la prueba:

Nombre de la prueba: (Nombre de la revisión del caso de prueba)

Prueba ejecutada por: (Nombre del probador que ejecutó la prueba)

Descripción: (Descripción breve de la revisión)

Fecha de la ejecución de la prueba:

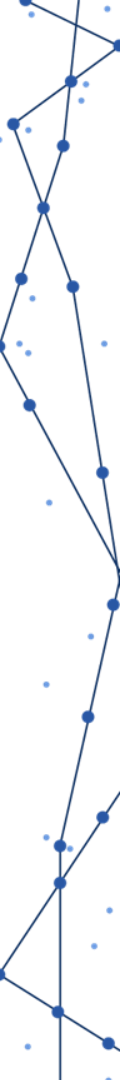
Pasos	Pasos de prueba	Datos de prueba	Resultado esperado	Resultado Actual	Estatus (Calificó/no calificó)	Observaciones
1	Elemento del módulo que se va a revisar	Descripción breve de su función	¿Qué es lo que se espera que haga el elemento?	¿Qué es lo que realmente hizo el elemento?	Si aprobó o no	
2						
3						



Después de pasar por el AOO

Diseño Orientado a Objetos (DOO)

Tarjetas CRC (Clase - Responsabilidad - Colaboración)





- Pruebas no convencionales
- Pruebas para aplicaciones Orientadas a Objetos:
 - Prueba de unidad en el contexto OO: A nivel clase y objeto encapsulado.
 - Prueba de integración en el contexto OO: Interacción entre clases (basadas en hilos y basadas en uso)
 - Prueba de validación en un contexto OO: Casos de uso. Acciones visibles al usuario y salida del sistema reconocida por él.



- Pruebas no convencionales
- Pruebas para aplicaciones Orientadas a Objetos:
 - Pruebas basadas en fallo.
 - Casos de prueba y jerarquía de clases.
 - Diseño de pruebas basadas en escenarios.
 - Pruebas aleatorias para clases OO.
 - Pruebas de partición a nivel clase.





- Pruebas no convencionales
 - Pruebas para aplicaciones para aplicaciones web:
Evalúan otros seis elementos:
 - Contenido
 - Interfaz
 - Navegación
 - Seguridad
 - Rendimiento (Carga (Peticiones) y estrés
(Carga*usuarios))

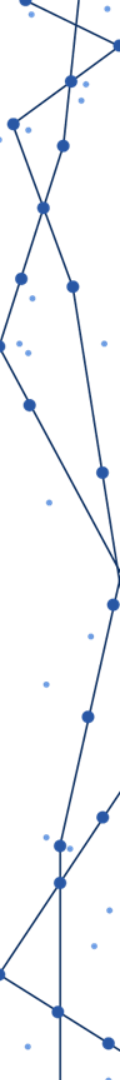


- Pruebas no convencionales
- Pruebas para aplicaciones para aplicaciones web:

OWASP

<https://owasp.org>

<https://www.veracode.com/security/owasp-top-10>



- Pruebas no convencionales
- Pruebas para aplicaciones para aplicaciones web:
OWASP, Generalidades:
 - ¿Existe la Seguridad de la información al 100%?
 - CID
 - Eslabón más débil de la seguridad
 - El software deberá responder a criterios de seguridad en cualquier ambiente, ya sea web o local.



- Pruebas no convencionales
 - Pruebas para aplicaciones para aplicaciones web:
- OWASP, Generalidades:
 - Vulnerabilidad de Sw: "Un fallo o hueco de seguridad detectado en algún programa o sistema informático que puede ser explotado"
 - Amenazas: Hackers, Virus, Ransomware, malware, etc.



- Pruebas no convencionales
- Pruebas para aplicaciones para aplicaciones web:
OWASP, Vulnerabilidades:
 - Buffer overflow
 - SQL Injection
 - HTML Injection
 - XSS
 - Errores de formato
 - Escalación de directorios
 - Errores en mecanismos de autenticación



- Pruebas no convencionales
- Pruebas para aplicaciones para aplicaciones web:

Desarrollo Seguro





- Pruebas no convencionales
 - Pruebas para aplicaciones para aplicaciones web:
 - Pruebas de la lógica de negocio
 - Pruebas de autenticación
 - Pruebas de autorización
 - Pruebas de gestión de sesiones
 - Pruebas de validación de datos
 - Pruebas de denegación de servicio
 - Pruebas de servicios web
 - Pruebas de AJAX



- Diseño de Casos de Prueba

Herramientas:

- Selenium: Automatiza pruebas y tareas web muy repetitivas. (Scripts).
- Editor de texto: Documentar y anexar imágenes.
- Camptasia Studio: para grabar la ejecución de tareas y sus resultados.

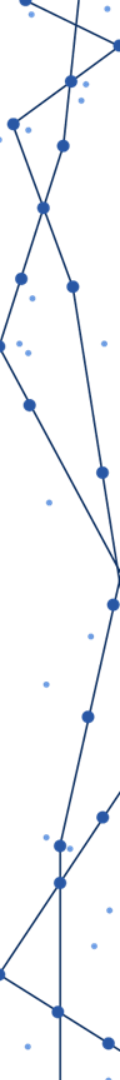


- Matriz de Prueba

Nombre del sistema:							
Fecha de revisión:				Encargado de la revisión:			
No.	Módulo	Descripción	Num. De pasos que aprobaron	Num. De pasos que NO aprobaron	Resultado esperado del módulo en general	Resultado real del módulo en general	Calificó / no calificó
1	(Nombre de los módulos que maneja el sistema)	(Una breve descripción de su funcionamiento)	(En base a la tabla del caso de prueba, poner el número de pasos que se aprobaron)	(En base a la tabla del caso de prueba, poner el número de pasos que no aprobaron)	Expectativa del funcionamiento del módulo	Realidad del funcionamiento del módulo	En base a la revisión realizada, calificarla aprobatoria o no aprobatoria
2							
3							
4							
5							

- Matriz de Prueba

Se puede ir armando con la identificación de casos de prueba, a continuación, se ejemplifica:





- Matriz de Prueba

6.0 Matriz de requisitos – casos de prueba

Módulo de la Secretaria

		Requisitos			
Pruebas					
	MSR1	MSR2	MSR3	MSR4	MSR5
MSP1	*				
MSP2		*			
MSP3			*		
MSP4				*	*

- Matriz de Prueba

Requisitos:

Módulo de la Secretaria

MSR1:	Registro de nuevos Asuntos.
MSR2:	Listado de Asuntos en trámite, por tipo de Asunto.
MSR3:	La secretaria solo puede llenar la sección de Datos Generales.
MSR4	Al turnar el asunto se envía un correo electrónico al TA/TUA.
MSR5:	Turnar el asunto al TA/TUA.

- Matriz de Prueba

Módulo del Titular del Área / Unidad Administrativa

[illegible]

- Matriz de Prueba

Requisitos

Módulo del Titular del Área / Unidad Administrativa

MTAR1: Registrar nuevos Asuntos.

MTAR2: Turnar el Asunto a departamento para el caso de Unidad Administrativa que tengan departamentos.

MTAR3: Asignar Jefe de Disciplina y notificación de correo electrónico.

MTAR4: Función Supervisión. Ver agenda de acciones de los Jefe de Disciplinas del área/UA, estado procesal de los asuntos y acciones tomadas. Capacidad de envío de correos a los Jefe de Disciplinas.

MTAR5: Estadísticas de su Área. Listado de los asuntos y estadísticas en trámite del área /unidad administrativa.

MTAR6: Desde la opción de Asuntos, mostrar listado de asuntos pendientes por asignar Jefe de Disciplina.

MTAR7: Opción de envío de mensajes sobre cada una de las acciones tomadas, y por tomar en el asunto. En caso de selección de no envío, se enviará solamente el mensaje de aviso de conclusión del asunto.

MTAR8: Capacidad de reasignar asuntos transferibles a otro Jefe de Disciplina.

- Matriz de Prueba

Módulo del Jefe de Departamento

		Requisitos	
Pruebas			
	MJDR1	MJDR2	MJDR3
MJDP1	*		
MJDP2		*	
MJDP3			*

- Matriz de Prueba

Módulo del Jefe de disciplina

					Requisitos						
Pruebas											
	MAR1	MAR2	MAR3	MAR4	MAR5	MAR6	MAR7	MAR8	MAR9	MAR10	MAR11
MAP1	*										
MAP2		*									
MAP3			*								
MAP4				*							
MAP5					*						
MAP6						*	*				

Resumen

Hoy aprendimos a ...

- **Diseñar Casos de Prueba.**
- **Diseñar Matrices de Prueba.**
- Prueba de caja blanca y sus tipos
- Estrategia de pruebas
- Pruebas de aplicaciones Orientadas o Objetos
- UML
- Casos de Uso
- Pruebas para aplicaciones Web



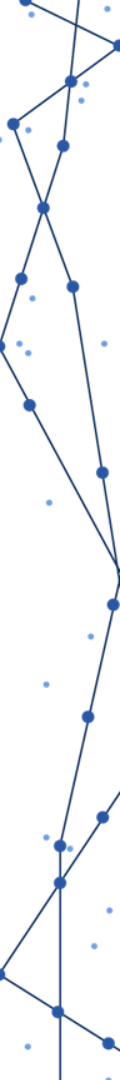
Referencias

- ## Bibliografía

Ingeniería en Desarrollo de Software. (2023). Temario 2, Visión General de las pruebas de Software, UMI, CDMX.

Ingeniería en Desarrollo de Software. (2023). Temario 3, Pruebas no convencionales, UMI, CDMX.

UML: Casos de Uso, Ingeniería del Software (Recuperado el 22 de 04 de 2023). Recuperado de:
<https://ingsoftwarekarlacevallos.wordpress.com/2015/06/04/uml-casos-de-uso/>





Ingenieria
Desarrollo de Software

