



TI área de desarrollo de software multiplataforma

4.- “B”

Proceso de testing en diferentes metodologías de desarrollo

Diego armando Dueñez villa 19170154

María Guadalupe Domínguez Vargas 19170093

Pedro Arturo de Santiago mijares 19170057

Ing. Abel Sosa Escobedo

06/10/2020

# PUNTOS A TRATAR

- Definición de una metodología de desarrollo
- Milestones
- Hitos
- Testing en RUP
- Testing en Agile
- User Stories
- Use Cases

# INTRODUCCIÓN

## (Definición de metodologías de desarrollo)

En ingeniería de software, es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Estas tienen como objetivo presentar un conjunto de técnicas tradicionales y modernas de modelado de sistemas que permitan desarrollar software de calidad, incluyendo descubrimientos de construcción y criterios de comparación de modelos de sistemas.



# MILESTONES

Milestones, es una metodología que aglutina nuestras experiencias obtenidas en situaciones reales de proyectos.

Mantenemos un proceso de mejora continua con el objetivo de asegurar la calidad de nuestros trabajos y procesos internos. Establecemos y dimensionamos equipos de trabajo altamente especializados, mediante los cuales podemos asegurar el cumplimiento de fechas y el compromiso con los objetivos y prioridades del proyecto.

# HITOS

Los hitos son una serie de etapas dentro de un mismo proyecto. Se determinan desde la planificación previa del mismo, se van revisando a medida que avanza nuestro trabajo y se pueden ir modificando según las necesidades del proyecto o cliente. Por eso decimos, que normalmente son partes indispensables en los proyectos que siguen una metodología ágil, cuyo requisito principal es la flexibilidad en la planificación.



# TESTING EN RUP

## (Definición de RUP)

RUP ó Proceso Unificado Racional (*Rational Unified Process*), es un proceso de desarrollo de software que utiliza el lenguaje UML para constituir la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.



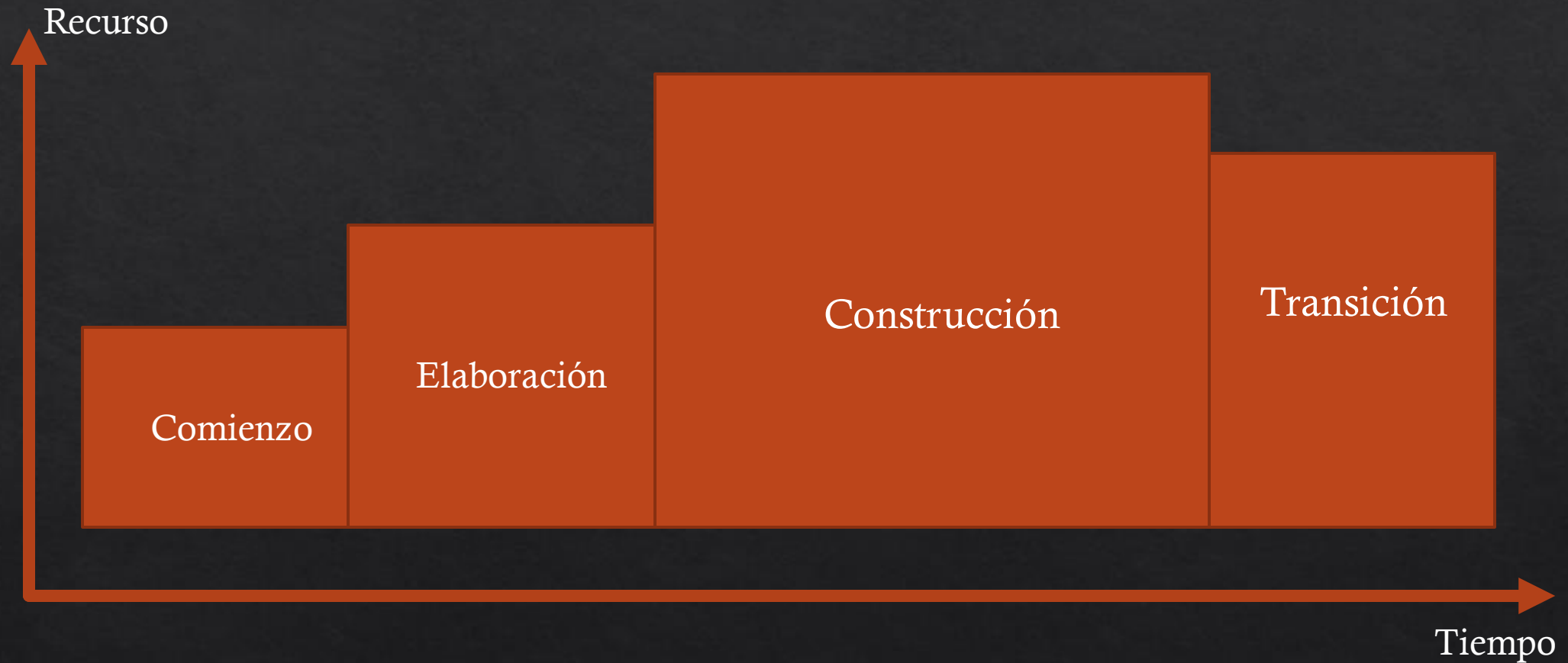
# TESTING EN RUP

## (Características del RUP)

- ✓ Forma disciplinada de asignar tareas y responsabilidades (Quien hace que, cuando y como).
- ✓ Pretende implementar las mejores practicas en ingeniería de software
- ✓ Desarrollo iterativo
- ✓ Administración de cambios
- ✓ Uso de arquitectura basada en componentes
- ✓ Control de cambios
- ✓ Modelo visual del software
- ✓ Verificación de la calidad del software

# TESTING EN RUP

## (Ciclo de vida del RUP)





# TESTING EN RUP

## (Elementos del RUP)

- ✓ **Actividades:** Son los procesos que se realizan en cada etapa.
- ✓ **Trabajadores:** Personas involucradas en cada actividad del proyecto.
- ✓ **Artefactos:** Herramientas empleadas para el desarrollo del proyecto (Documentos, modelos, diagramas).

# TESTING EN RUP

## (Proceso de pruebas)

Las pruebas en RUP se inician desde la fase de **Elaboración** verificando el cumplimiento de los prototipos respecto a los requerimientos de los usuarios, luego en la fase de **Construcción** con más énfasis cuando se van integrando todos los módulos o componentes del sistema y se deben realizar las pruebas integrales.



# TESTING EN RUP

## (Pruebas que se suelen realizar)

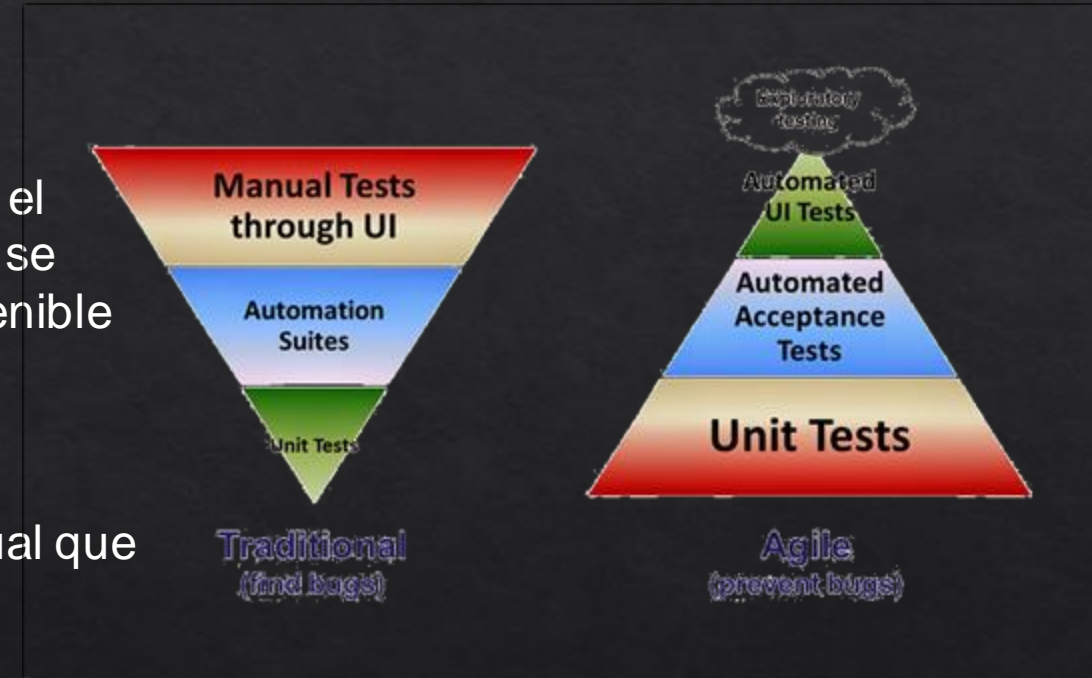
- ✓ **Pruebas de Funcionalidad:** Orientadas a determinar el cumplimiento de las funcionalidades del sistema.
- ✓ **Pruebas de integridad:** Orientadas a probar el sistema en su conjunto y evaluar si los módulos y componentes operan bien estando interrelacionados.
- ✓ **Pruebas de Sobrecarga:** Orientadas a probar fundamentalmente el desempeño de la base de datos, la velocidad de los algoritmos de búsqueda de datos y las funciones de ordenamiento de la base de datos.
- ✓ **Pruebas de Tensión:** Orientadas a probar los programas de manera simultánea desde varias terminales.
- ✓ **Pruebas de ergonomía del sistema:** Orientadas a evaluar la facilidad de uso de los programas, su amigabilidad, su personalización, etc.

# TESTING EN AGILE

Agile Testing es una práctica de pruebas de software que sigue los principios del desarrollo ágil de software.

Involucra a todos los miembros de un equipo ágil multifuncional, en el cual el rol del tester es el de un experto multifuncional, garante que se entregue el valor de negocio deseado por el cliente a un ritmo sostenible y continuo.

Las metodologías ágiles no ven al software testing como una fase separada, sino como parte integral del Desarrollo de software al igual que la programación.



# Principios del Agile Testing

- ◊ El Testing no es una fase
- ◊ El Testing hace avanzar el proyecto
- ◊ Todo el equipo realiza pruebas
- ◊ Reducir el tiempo para recibir retroalimentación
  - ◊ Código limpio:
- ◊ Reducir la documentación de pruebas
  - ◊ Guiado por pruebas



# User Stories

- ❖ Las Historias de Usuario son un elemento básico para aplicar metodologías Ágiles y especialmente para poder aplicar SCRUM.
- ❖ Su simpleza hace de esta técnica una gran herramienta para poder tratar casi todos los aspectos necesarios para la creación de productos, especialmente los de software. Y todo se basa en una regla de palabras muy curiosa:

COMO <rol>  
QUIERO <evento>  
PARA <funcionalidad>

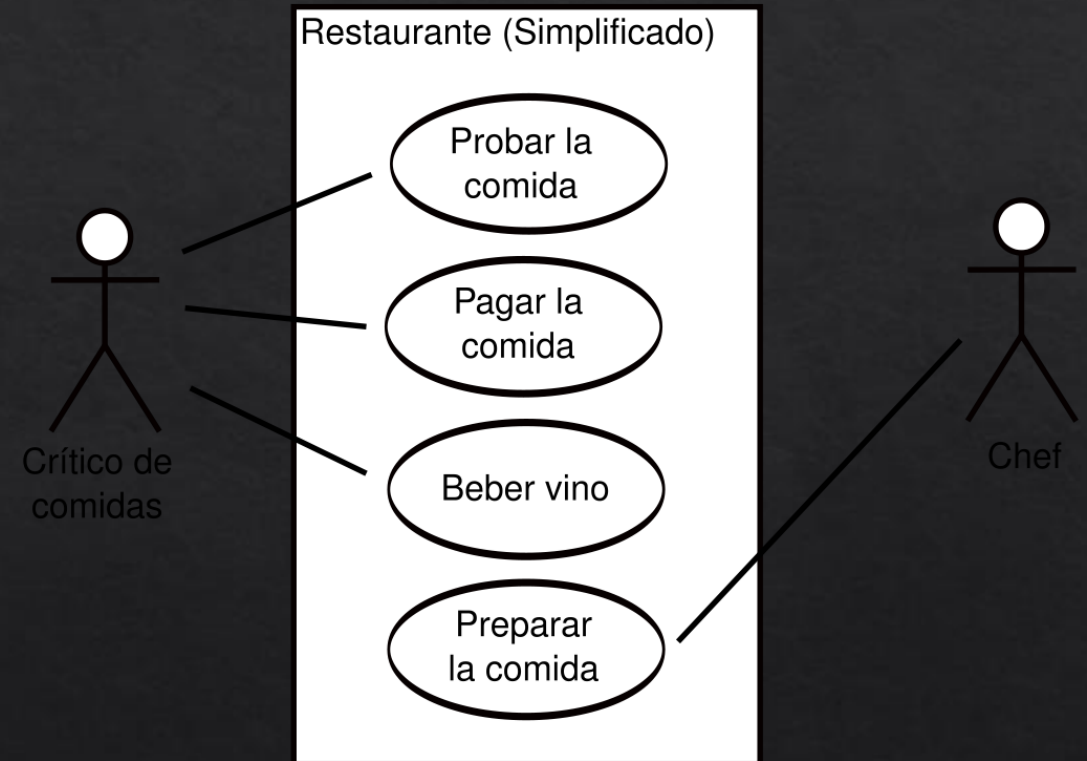
COMO us de la  
tienda on line  
QUIERO botón ordenar  
por precio  
PARA escoger el gel  
más barato y comprar

- ❖ Una historia de usuario es una nota que captura lo que un usuario hace o necesita hacer como parte de su trabajo. Cada historia de usuario consta de una breve descripción escrita desde el punto de vista del usuario, con lenguaje natural. A diferencia de la captura de requisitos tradicional, User Story se centra en lo que el usuario necesita en lugar de en lo que el sistema debería ofrecer. Esto deja espacio para una mayor discusión de las soluciones y el resultado de un sistema que realmente puede encajar en el flujo de trabajo comercial de los clientes, resolviendo sus problemas operativos y, lo más importante, agregando valor a la organización.

COMO DESARROLLADOR  
QUIERO UNA HERRAMIENTA  
PARA RECOGER LOS  
REQUISITOS DEL SW  
FÁCIL Y BIEN

# Use Cases

- ◆ El caso de uso es una buena herramienta para estimular a que los usuarios potenciales hablen, de un sistema, desde sus propios puntos de vista. No siempre esto es una tarea fácil. La idea es involucrar a los usuarios en las etapas iniciales de análisis y diseño del sistema. Esto aumenta la probabilidad de que el sistema sea de mayor utilidad en lugar de ser un conjunto de funcionalidades incomprensibles e inmanejables por los usuarios finales.



- ◆ Si bien los casos de uso son historias estilizadas en la manera en que un usuario final desempeña uno o varios papeles posibles, así como interactúa con un sistema en un conjunto específico con las circunstancias.
- ◆ Estas historias pueden ser narradas en:
  - ◆ Esquema de tareas
  - ◆ Interacciones
  - ◆ Descripciones basadas en plantillas
  - ◆ Representadas por diagramas.



# CONCLUSION

A lo largo de la materia hemos sabido que es parte muy importante implementar a los proyectos de software la parte del proceso de testing, es una buena forma de implementar el tiempo de desarrollo que tenemos para que sean mínimas las posibilidades de fallas en nuestro programa.

Como vimos en la presentación, el testing de alguna u otra manera está presente y se acopla de alguna forma a la metodología que se use, en este caso vimos algunas y las maneras en que se testea en estas mismas.

A un alto nivel, el testing en software es necesario para detectar los errores y para probar si el software cumple con los requisitos del cliente. Esto ayuda al equipo de desarrollo a corregir los errores y entregar un producto de buena calidad.