

Ingeniería de Software II - XP

MARIO ROSSAINZ LÓPEZ

OTOÑO 2022

NRC: 10838

PROGRAMACIÓN EXTREMA (XP)

- Metodología de desarrollo ágil formulada por Kent Beck en 1999.
- Pone más énfasis en la *adaptabilidad* que en la *previsibilidad*.
- Considera que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable.
- Utiliza como enfoque preferido de programación a la Orientación a Objetos

PROGRAMACIÓN EXTREMA (XP)

Valores:

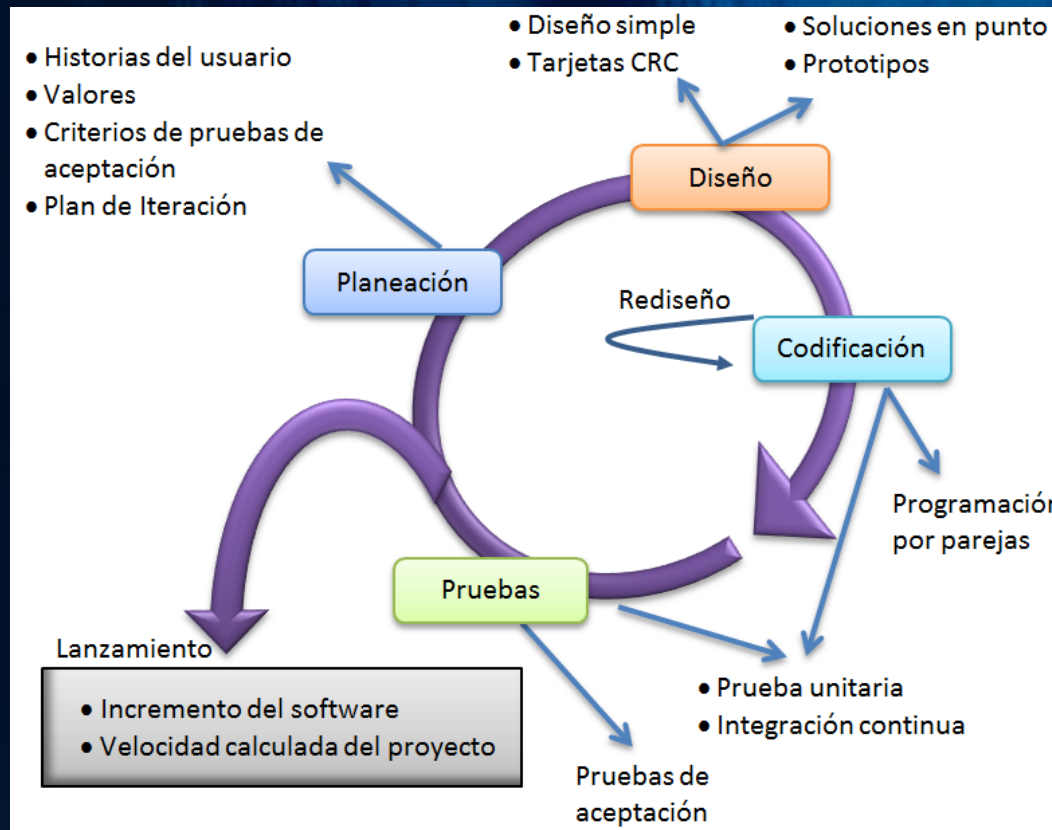
- Comunicación
- Retroalimentación
- Simplicidad
- Coraje

Principios:

- Retroalimentación rápida
- Cambio incremental
- Trabajo de calidad
- Simplicidad

PROGRAMACIÓN EXTREMA (XP)

La XP abarca un conjunto de reglas y prácticas que ocurren en el contexto de 4 actividades del marco de trabajo: Planeación, Diseño, Codificación y Pruebas.



PROGRAMACIÓN EXTREMA (XP)

Planeación:

- Comienza creando una serie de historias que describen las características y la funcionalidad requeridas para el software que se construirá. Cada historia las escribe el cliente y se coloca en una carta índice.
- El cliente le asigna un valor a la historia basándose en los valores generales del negocio.
- Los miembros del equipo PE evalúan las historias y le asignan un costo, el cuál se mide en semanas de desarrollo. Si la historia requiere mas de tres semanas se solicita una reformulación.
- Los clientes y el equipo PE trabajan juntos para decidir cómo agrupar las historias hacia el próximo lanzamiento. Se debe generar un compromiso básico de las historias que se desarrollaran en un incremento.

PROGRAMACIÓN EXTREMA (XP)

Planeación:

- Una vez establecido el compromiso básico, el equipo PE ordena las historias que se desarrollaran de una de las siguientes tres maneras:
 1. Todas las historias serán implementadas de un modo inmediato (dentro de pocas semanas).
 2. Las historias con valor mas alto se moverán en el programa y se implementaran al principio.
 3. Las historias más riesgosas se moverán dentro del programa y se implementarán al principio.
- Después de que se realiza el primer lanzamiento el equipo de PE calcula la velocidad del proyecto (La velocidad del proyecto es el número de historias implementadas en un lanzamiento).

PROGRAMACIÓN EXTREMA (XP)

Planeación:

- La velocidad del proyecto puede utilizarse para:
 1. Estimar fechas de entrega y lanzamientos subsecuentes.
 2. Determinar si se ha hecho un compromiso excesivo en algunas de las historias de todo el proyecto de desarrollo.
- Es importante tener claro que, el cliente puede agregar historias, cambiar el valor de la historia existente, dividir historias o eliminarlas. Por lo que el equipo modifica los lanzamientos restantes y sus planes.

PROGRAMACIÓN EXTREMA (XP)

Diseño:

- Esta actividad sigue de manera rigurosa el principio de mantener las cosas simples. Se prefiere simplicidad a complejidad.
- El diseño es una guía de implementación para una historia como esta escrita, ni más ni menos.
- La PE apoya el uso de las tarjetas CRC como un mecanismo efectivo para pensar en el software en un contexto orientado a objetos. Estas tarjetas son el único producto del diseño.
- Este modelo de clases Responsabilidad-Colaborador (CRC) proporciona un medio simple para identificar y organizar las clases relevantes para los requisitos del sistema o producto.

PROGRAMACIÓN EXTREMA (XP)

Diseño:

Clase: Nombre de la clase	
Lista de Superclases	
Lista de Subclases	
Responsabilidad	Colaboración

TARJETA DE CLASE

PROGRAMACIÓN EXTREMA (XP)

Diseño:

- Si el diseño de una historia se encuentra difícil, la PE recomienda la creación inmediata de un prototipo operacional de esa porción del diseño.
- La PE apoya la refabricación (refactoring), una técnica de construcción que también es de diseño. Esto ya que el diseño se considera una fase que puede y debe modificarse de manera continua a medida que prosigue la construcción.

PROGRAMACIÓN EXTREMA (XP)

Codificación:

- La PE recomienda que después de diseñar las historias y realizar el trabajo de diseño preliminar el equipo no debe moverse hacia la codificación, sino que debe desarrollar una serie de pruebas de unidad.
- Una vez desarrollada las pruebas el desarrollador puede centrarse en implementar lo necesario para cumplirlas.
- Uno de los aspectos importantes de la codificación es la **PROGRAMACION EN PAREJAS**. PE recomienda que dos personas trabajen juntas en una estación de trabajo creando una historia (se sigue el concepto dos cabezas piensan mejor que una).
- También existe una **integración continua**, ya que cada vez que se termina de codificar una historia se debe integrar esta parte con las realizadas por los demás.

PROGRAMACIÓN EXTREMA (XP)

Codificación:

Algunas buenas prácticas utilizadas en esta etapa son:

- ✓ **Programación por Pares (Pair Programming):** El código debe ser desarrollado por parejas que trabajen sobre la misma máquina. Esto que, a simple vista parece un gasto de recursos, consigue por un lado que el código desarrollado normalmente sea de mayor calidad (2 pares de ojos ven mejor que uno) y, por otro, que los desarrolladores trabajen todo el tiempo.
- ✓ **Refactoring:** Una vez que pase los test un método se revisa para ver si se puede mejorar. Se refactoriza en todas las etapas del desarrollo, no solo al final.

PROGRAMACIÓN EXTREMA (XP)

Codificación:

Algunas buenas prácticas utilizadas en esta etapa son:

- ✓ **Propiedad colectiva del código (Collective Ownership):** Todo es responsabilidad de todos. Ninguna línea de código es conocida solo por un desarrollador.
- ✓ **Estándares de Codificación:** Se siguen una serie de estándares a la hora de realizar el código de tal forma que cualquiera lea lo mismo en el mismo fragmento. Esto ayuda a que cualquier desarrollador pueda seguir el trabajo de otro cuando se rotan las parejas y a la transferencia de conocimiento dentro del equipo de desarrollo.

PROGRAMACIÓN EXTREMA (XP)

Codificación:

Algunas buenas prácticas utilizadas en esta etapa son:

- ✓ **Integración Continua:** El código debe ser probado e integrado cada poco tiempo con lo que los errores de codificación y de integración se detectan y se corrigen mucho antes.

PROGRAMACIÓN EXTREMA (XP)

Pruebas:

- Las pruebas de unidad que se crean deben implementarse con un marco de trabajo que permita automatizarlas. Esto apoya una estrategia de regresión de prueba.
- Se recomienda que las pruebas de integración y de sistemas se vayan realizando a medida que se realizan un conjunto pequeño de pruebas de unidades. Esto se refrenda con la siguiente frase:

"Arreglar problemas pequeños cada pocas horas toma menos tiempo que arreglar problemas enormes justo antes de la fecha límite."

- Las pruebas de aceptación o cliente, las especifica el cliente. Se realizan en base a las historias de usuario.

PROGRAMACIÓN EXTREMA (XP)

Pruebas:

Algunas buenas prácticas utilizadas en esta etapa son:

- **Pruebas Automatizadas (testing):** Programación orientada por pruebas. El cliente realiza test funcionales y el desarrollador realiza tests por cada característica nueva.

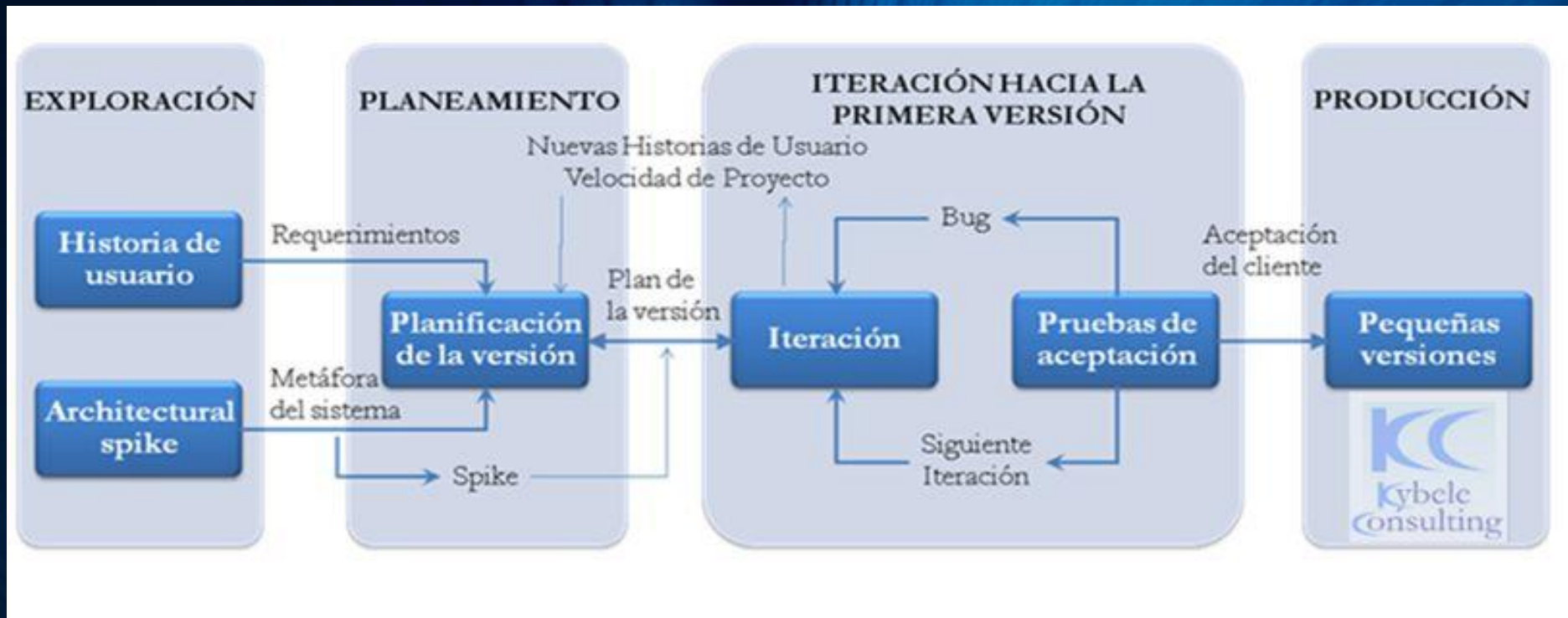
PROGRAMACIÓN EXTREMA (XP)

Prácticas que involucran a todo el desarrollo.

- Cuarenta horas semanales: Semanas de 40 horas de trabajo al 100% para todo el mundo del equipo. El trabajo del día a día está dirigido a maximizar la productividad, con lo que 40 horas semanales se ve más que suficiente.
- Cliente siempre disponible: Es necesario que el usuario del sistema esté presente para resolver dudas y definir propiedades a más baja escala.

PROGRAMACIÓN EXTREMA (XP)

Ciclo de vida de un proyecto XP:



PROGRAMACIÓN EXTREMA (XP)

Historias de usuario:

- Es una representación de un requisito utilizando el lenguaje común del usuario. Se utilizan para especificar los requisitos.
- Cada historia de usuario debe ser limitada y escrita por los clientes

PROGRAMACIÓN EXTREMA (XP)

Historias de usuario [Características]:

- Independientes unas de otras
- Negociables (alcance claro)
- Valoradas por los usuarios (no por el desarrollador)
- Estimables en el tiempo
- Pequeñas
- Verificables

PROGRAMACIÓN EXTREMA (XP)

Historias de usuario [Características]:



Una historia de usuario es una narración que describe una funcionalidad del sistema que tiene **valor** para un usuario

Se recogen en unas sencillas tarjetas de forma esquemática y en un lenguaje claro y preciso

<i>Aprobación de nuevos usuarios</i>	
<i>¿quién?</i>	<i>Yo como administrador del foro quisiera poder aceptar o rechazar los nuevos usuarios registrados para así poder evitar que el foro se llene de spammers</i>
<i>¿por qué?</i>	
	<i>¿qué?</i>

PROGRAMACIÓN EXTREMA (XP)

Historias de usuario [Ejemplos]:

Historia: Agregar comentarios

Como: Lector del Blog

Quiero: adicionar comentarios a las entradas y recibir alertas cuando otros hagan comentarios

Para: mantenerme en contacto con los demás usuarios del blog

3

Historia: Responder a comentarios

Como: Lector del Blog

Quiero: adicionar comentarios a las entradas y responder a comentarios de otros lectores

Para: mantenerme en contacto con los demás usuarios del blog

3

PROGRAMACIÓN EXTREMA (XP)

Historias de usuario [Ejemplos]:

Historia de Usuario	
Número: 3	Nombre: Tabla de Magnitudes del Balance.
Usuario: Especialista	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1.5
Riesgo en Desarrollo: Alta	Puntos Reales: 1.2
Descripción: A partir de las mediciones anuales insertadas en la rejilla de datos de la ventana de trabajo se calcularán las magnitudes del balance y se mostrarán en forma de tabla, agrupadas y promediadas por años. La tabla de magnitudes se mostrará en una nueva ventana de trabajo que presentará una barra de herramientas con las funciones de la categoría Impresión, Gráficos y Tabla de Probabilidades.	
Observaciones: La Tabla de Magnitudes del Balance podrá generarse a partir de las ventanas de trabajo Método de Balance e Información General de Niveles, utilizando los valores de la variable Nivel y Precipitaciones. Se podrá calcular los valores de las magnitudes del balance del Método de Balance a diferentes probabilidades.	

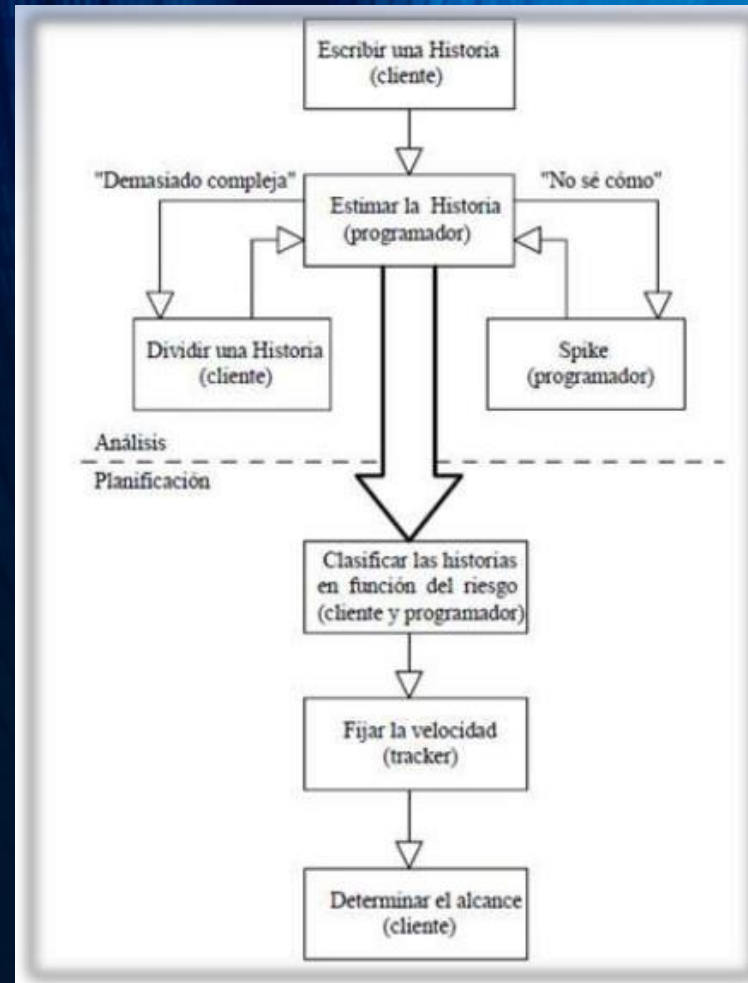
PROGRAMACIÓN EXTREMA (XP)

Historias de usuario [Ejemplos]:

Historia de Usuario	
Número: 4	Usuario: Administrador, Usuario.
Nombre: Gestionar AT3	
Prioridad en Negocio: Alta	Puntos Estimados: 1.2
Riesgos en Desarrollo: Alta	Iteración Asignada: 1
Descripción: Se podrá insertar, modificar, eliminar y buscar los datos del control de materiales básicos entregados según la fecha de entrega por trabajador.	
Observaciones:	

PROGRAMACIÓN EXTREMA (XP)

Historias de usuario [Ejemplos]:



PROGRAMACIÓN EXTREMA (XP)

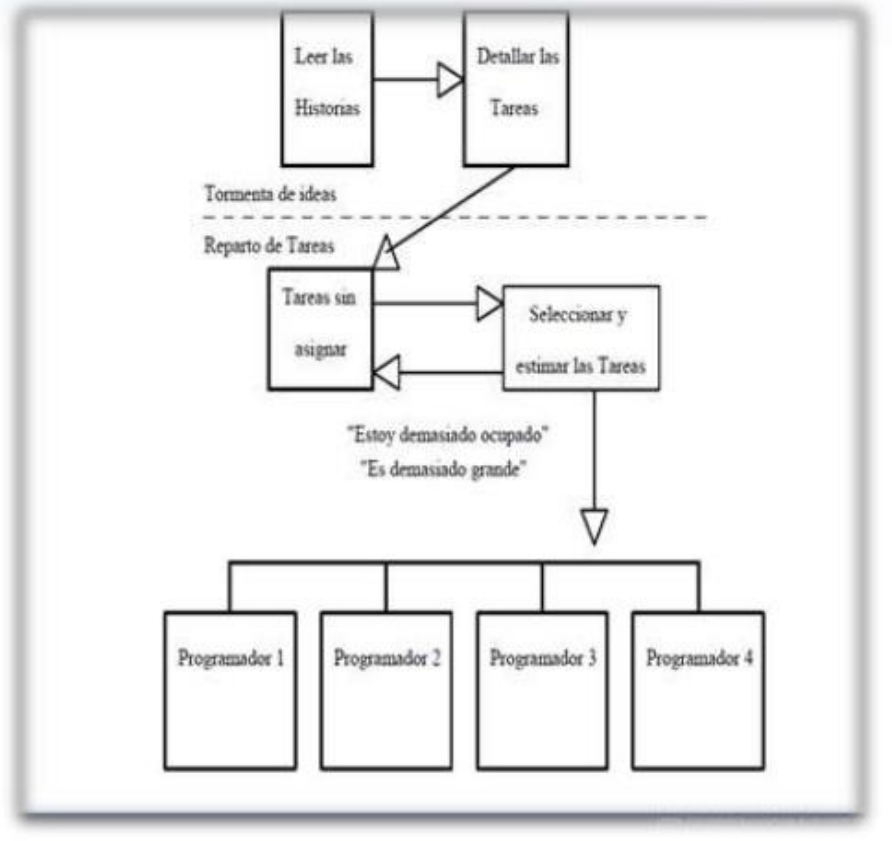
Historias de usuario [Ejemplos]:

Calendario de Trabajo	
Semana 3-12 de Mayo	
Martes 4	Negociación de requisitos.
Miércoles 4 a Martes 11	Desarrollo de requisitos R1 y R2
Miércoles 12	Entrega y Retroalimentación con el Usuario.
Semana 13-19 de Mayo	
Jueves 13 a Martes 18	Desarrollo paralelo del R3 y R4.
Miércoles 19	Entrega y Retroalimentación con el Usuario.
Semana 20-26 de Mayo	
Jueves 20 a Martes 25	Desarrollo paralelo del R3 y R4.
Miércoles 26	Entrega.

PROGRAMACIÓN EXTREMA (XP)

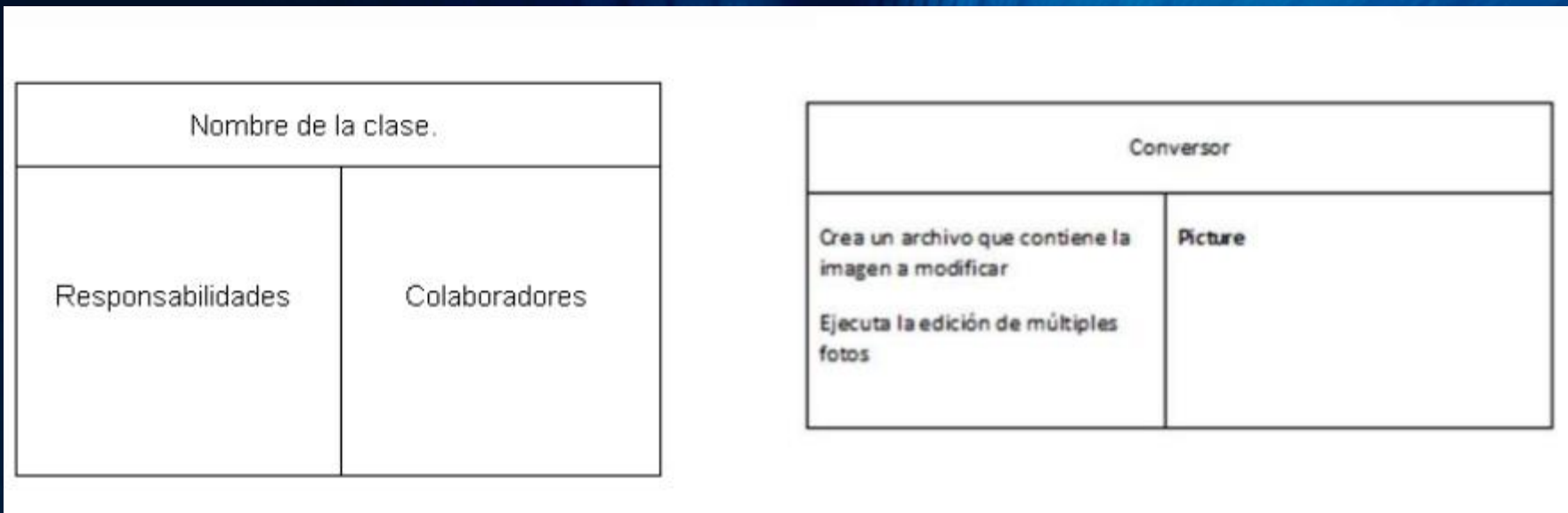
Historias de usuario [Ejemplos]:

Así puede verse un plan de iteración:



PROGRAMACIÓN EXTREMA (XP)

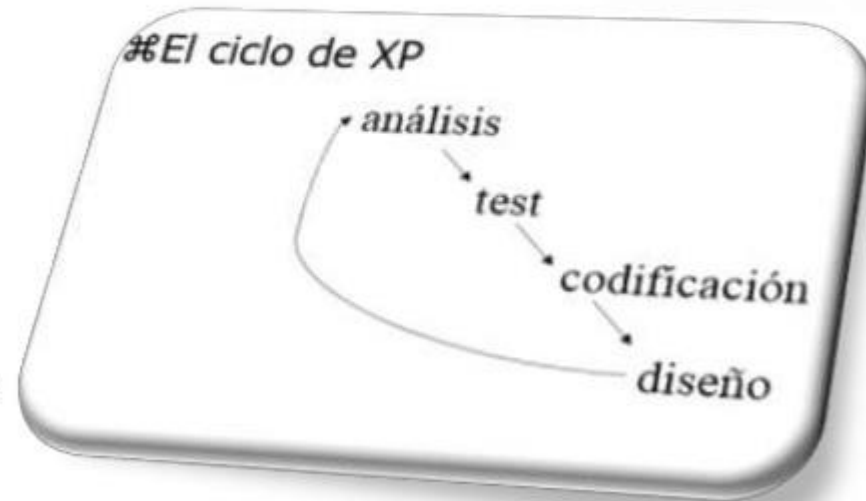
Diseño:



PROGRAMACIÓN EXTREMA (XP)

Codificación:

- Disponibilidad del CLIENTE
- Unidades de prueba o test
- Programación pareja
- Integración del código
- Frecuencia en la integración del código
- El código es propiedad de todos:
- Dejar la optimizaciones para el final
- No a las horas extras: 40 horas semanales



PROGRAMACIÓN EXTREMA (XP)

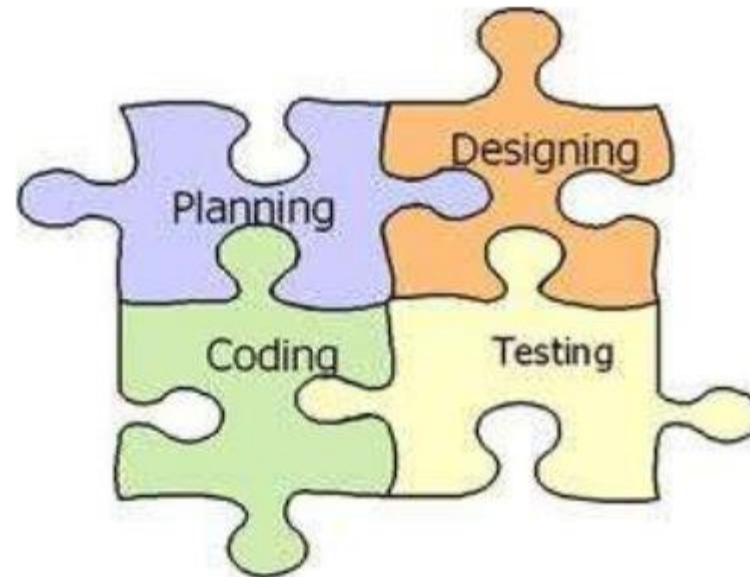
Pruebas:

Unidades de test o pruebas: Pilar básico

Implantación: El código será implantado cuando supere sus correspondientes unidades de test.

Protección contra fallos: Solución, un test

Pruebas de aceptación: Evaluación del cliente



PROGRAMACIÓN EXTREMA (XP)

Roles de la XP:

- Programador
- Cliente
- Encargado de Pruebas (tester)
- Encargado de Seguimiento (tracker)
- Entrenador (coach)
- Gestor (big boss)