

**UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE SEDE  
SANTO DOMINGO**

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN - DCCO-SS**

**CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA  
INFORMACIÓN**

**PERIODO** : Noviembre 2023 – Marzo 2024

**ASIGNATURA** : Metodología de Desarrollo de Software

**TEMA** : Programación Extrema

**NOMBRES** : Esteban Larco, Aldo Saula, Eduardo Camino

**SEMESTRE** : Tercero

**DOCENTE** : Ing. Javier Cevallos Farías.

**FECHA DE ENTREGA** : 17/1/2024

**SANTO DOMINGO - ECUADOR**

**2023**

# ÍNDICE

	<b>1</b>
<b>1. Introducción</b>	<b>3</b>
<b>2. Sistemas de Objetivos</b>	<b>3</b>
<b>Objetivo General:</b>	<b>4</b>
<b>Objetivos Específicos:</b>	<b>4</b>
<b>3. Desarrollo</b>	<b>4</b>
<b>¿Qué es?</b>	<b>4</b>
<b>Valores de la Metodología XP</b>	<b>4</b>
<b>Principios de XP</b>	<b>5</b>
<b>Metodologia XP: Fases</b>	<b>6</b>
<b>Prácticas de XP</b>	<b>7</b>
<b>Roles de XP</b>	<b>8</b>
<b>Gestión de proyectos</b>	<b>8</b>
<b>XP intensa pero efectiva</b>	<b>9</b>
<b>4. Conclusiones</b>	<b>9</b>
<b>5. Recomendaciones</b>	<b>10</b>
<b>6. Bibliografía/ Referencias</b>	<b>10</b>

## 1. Introducción

La Programación Extrema emerge como una metodología ágil e innovadora, enfocada en la creación de software de alta calidad y la adaptación a requisitos cambiantes. Siguiendo el objetivo general de proporcionar una comprensión integral de la XP, examinaremos de manera detallada los valores que sustentan esta metodología, como la comunicación, la sencillez, el coraje, el respeto y la retroalimentación. No solo nos quedaremos en la teoría, sino que también explicaremos las fases concretas de la Metodología XP, desde la planificación hasta el lanzamiento, destacando cómo cada etapa contribuye al desarrollo iterativo e incremental. Adicionalmente, revisaremos las prácticas clave de XP, como la programación en pareja, la propiedad colectiva del código y la integración continua, que forman la esencia de esta metodología.

" Con esta premisa en mente, embarquemos en un viaje a través de la Programación Extrema, explorando sus conceptos clave y desentrañando el enfoque riguroso pero efectivo que define esta metodología ágil."

Palabras clave: Programación Extrema (XP), Metodología Ágil, Valores de XP, Principios rectores, Fases de implementación, Prácticas clave de XP, Desarrollo iterativo e incremental, Comunicación efectiva, Sencillez en el diseño, Retroalimentación continua, Coraje en el cambio, Respeto en el trabajo en equipo, Propiedad colectiva del código, Integración continua, Roles de XP, Cliente (Customer), Programador (Developer), Entrenador (Coach), Rastreador (Tracker), Probador (Tester), Pronosticador (Doomsayer), Gestión de proyectos en XP, Historias de usuarios, Ciclos semanales y trimestrales, Slack en la planificación, Diseño sencillo, Dirección de desarrollo de pruebas, Refactorización, Programación en pareja, Casos de éxito en XP.

## 2. Sistemas de Objetivos

**Objetivo General:**

Proporcionar una comprensión integral de la metodología de programación extrema (XP), destacando sus principios fundamentales, procesos clave y beneficios, con el fin de informar a los lectores sobre este enfoque ágil de desarrollo de software..

### **Objetivos Específicos:**

- Explorar los Principios de la Programación Extrema (XP).
- Detallar las Prácticas Clave de la Programación Extrema.
- Examinar Casos de Éxito y Desafíos de Implementación de XP.

## **3. Desarrollo**

### **¿Qué es?**

La Programación Extrema es un método de desarrollo de software incluido en la denominada Metodología Ágil. XP se basa en valores, principios y prácticas y tiene como objetivo permitir que equipos pequeños y medianos creen software de alta calidad y se adapten a requisitos cambiantes y en evolución.

### **Valores de la Metodología XP**

#### **Comunicación**

Para garantizar que los programadores comprendan las necesidades de los clientes y que los propios clientes comprendan las capacidades, la arquitectura y los objetivos del sistema, Extreme Programming proporciona conversaciones personales y directas entre las partes.

#### **Sencillez**

Para evitar desperdicios, reducir costos y tiempo, y mantener el diseño y la funcionalidad lo más fáciles de usar posible, XP intenta priorizar lo que realmente se necesita en el proyecto.

#### **Comentario**

Para garantizar ajustes más rápidos y precisos, es esencial contar con retroalimentación continua y a corto plazo y retroalimentación a los profesionales en las primeras etapas del proceso.

#### **Coraje**

Estar dispuesto a cambiar, manejar el fracaso, aceptar comentarios, sugerir mejoras y ser capaz de decir “no” cuando sea necesario significa confiar en el proceso. En su libro Extreme Programming Guide, el autor Kent Beck define el coraje como "actuar eficazmente frente al miedo".

#### **Respeto**

El trabajo en equipo es una de las condiciones de XP y requiere que los miembros se respeten entre sí,

aceptan sugerencias, cooperen entre sí y valoren las buenas relaciones.

### Retroalimentación

La retroalimentación en las metodologías de desarrollo de software más tradicionales, tipo cascada, a menudo es «demasiado pequeña, demasiado tarde».

XP, sin embargo, adopta el cambio y los equipos de XP se esfuerzan por recibir una retroalimentación temprana y constante. Si es necesario corregir el rumbo, los XPers quieren saberlo lo antes posible.

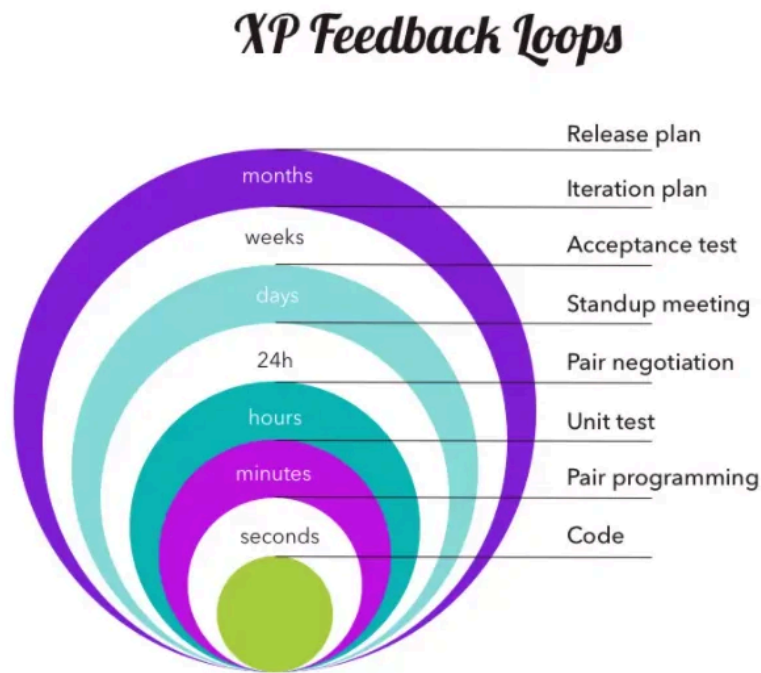


Image src: [extremeprogramming.org](http://extremeprogramming.org)

### Principios de XP

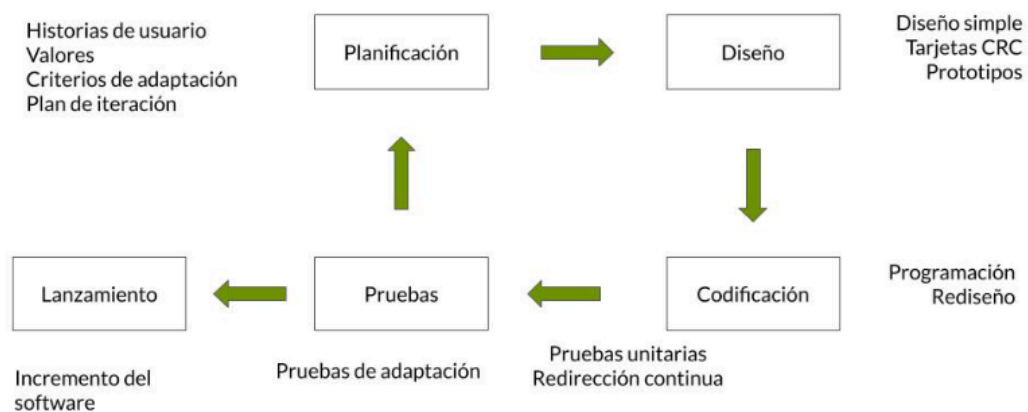
Las directrices proporcionan una orientación más específica que valor. Se trata de principios rectores que clarifican los valores y los hacen más claros y menos ambiguos.



Image src: [extremeprogramming.org](http://extremeprogramming.org)

## Metodología XP: Fases

### Metodología XP o Programación Extrema



Sinnaps

### Planificación

A partir de la identificación de historias de usuarios, se priorizan y se dividen en versiones pequeñas. Se mejorará el diseño. Aproximadamente cada dos semanas, debería tener un software útil y funcional listo para probar y lanzar.

### Diseño

En esta etapa intentaremos trabajar con código simple, haciendo lo mínimo necesario para que funcione.

Conseguiremos un prototipo. Además, se creará una etiqueta CRC (Responsabilidad de clase de colaboración) para el diseño de software orientado a objetos.

### **Codificación**

La programación se realiza aquí con ambas manos, en parejas frente a una computadora. A veces incluso intercambian parejas. De esta manera nos aseguramos de crear un código más universal con el que cualquier otro desarrollador pueda trabajar y comprender. Y parecerá que alguien lo hizo. Con ello se logrará un programa organizado y planificado.

### **Verificar**

Las pruebas automatizadas deben ejecutarse continuamente. Dado que suelen ser proyectos a corto plazo, las pruebas automatizadas y continuas son importantes. Además, los clientes pueden realizar sus propias pruebas, proponer nuevas pruebas y validar versiones menores.

### **Lanzamiento**

Si llegamos a este punto, habremos probado con éxito todas las historias de usuario o versiones pequeñas requeridas por el cliente. Disponemos de software útil y podemos implementarlo en el producto.

### **Prácticas de XP**

Extreme Programming, además de valores y principios, también ofrece buenas prácticas en la planificación, organización, comunicación y desarrollo de software que, combinadas con los valores y principios, crean una excelente cultura de equipo.

### **El juego de planificación**

La idea principal detrás de este enfoque es dividir las responsabilidades de planificación entre el equipo y el cliente, lo que significa que requiere la participación activa del cliente en el proceso de planificación. Se basa en un principio simple: los clientes suelen tener información completa sobre los costos del proyecto y los equipos saben todo sobre sus costos.

### **Lanzamientos pequeños**

El equipo de desarrollo puede lanzar periódicamente versiones iterativas del sistema para los clientes. Metáfora. Podemos explicar el diseño del sistema a gente nueva usando metáforas en lugar de pedirles que lean muchos documentos.

### **Diseño sencillo**

La regla es, como su nombre indica, que sea sencillo.

### **Dirección de desarrollo de pruebas**

Es un enfoque evolutivo para el desarrollo de software, que combina dos prácticas que nos permiten crear código de alta calidad, pensar en la arquitectura del software que queremos desarrollar, escribir pruebas

experimentales primero y luego mejorarlo mediante la refactorización.

### **Refactorización**

mejorar el diseño del código existente sin cambiar su comportamiento.

### **Programación en pareja**

dos programadores trabajando en parejas en la misma máquina resolverán el mismo problema.

### **Propiedad colectiva del código**

Ningún miembro de todo el equipo posee una pieza específica de código fuente, el código pertenece a todos.

### **Integración Continua**

Método diseñado específicamente para construir o integrar todas las etapas de desarrollo, identificar errores y eliminarlos durante el desarrollo, reduciendo así los tiempos de respuesta y mejorando la calidad del software creado.

### **Historias de usuarios**

La historia es un breve documento de texto escrito por el cliente que describe en términos sencillos por qué necesita el software.

### **Roles de XP**

Un equipo de XP incluye seis roles:

**El cliente (Customer):** Es la persona responsable de escribir historias de usuarios, establecer prioridades y formular la cartera de productos.

**El programador (Developer):** Es un desarrollador normal, que escribe el código y realiza la totalidad de las tareas del proyecto.

**El entrenador (Coach):** Es la persona que vigila el trabajo del equipo, lo controla y enseña a sus miembros a implementar las prácticas más efectivas.

**El rastreador (Tracker):** Es la persona cuya tarea principal es monitorear el progreso del desarrollo del software y detectar todos los problemas en él.

**El probador (Tester):** Es el miembro del equipo responsable de la prueba del producto. La calidad del producto final depende en gran medida de su trabajo.

**El pronosticador (Doomsayer):** Es la persona que rastrea los riesgos del proyecto y advierte al equipo sobre ellos.

### **Gestión de proyectos**

#### **Historias**

Los requisitos del usuario se escriben en un formato conocido como historias de usuario. Una historia de usuario tiene un nombre corto y descriptivo y también una breve descripción de lo que se va a implementar.



## **Slack**

Cuando se planifica un ciclo, se añaden tareas menores que el equipo puede abandonar en caso de que lo necesite. Siempre se pueden añadir más historias si el equipo se excede.

## **Ciclos (trimestrales y semanales)**

El desarrollo en XP ocurre en dos ciclos principales: el ciclo semanal y el ciclo trimestral. Más sobre esto en la siguiente sección.

## **Reuniones clave, ciclos, cadencias de entrega**

El desarrollo en XP funciona en dos ciclos principales: el ciclo semanal y el ciclo trimestral. Inicialmente, Kent Beck recomendaba un ciclo de dos semanas, pero lo cambió en la segunda edición de su libro.

## **Ciclo semanal**

El ciclo semanal es el «pulso» de un proyecto XP. El ciclo comienza con una reunión en la que el cliente elige qué historias quiere implementar durante la semana. Además, el equipo revisa su trabajo, incluyendo el progreso de la semana pasada, y piensa en cómo mejorar su proceso.

## **Ciclo trimestral**

Cada trimestre, el equipo reflexiona e identifica oportunidades de mejora en su proceso. El cliente elige uno o más temas para ese trimestre, junto con las historias de estos temas.

## **XP intensa pero efectiva**

A esta altura, probablemente te hayas dado cuenta de que la programación extrema es, en esencia, extrema. El proceso es muy riguroso y altamente estructurado, pero los resultados realmente valen la pena. El proceso de desarrollo único de XP que incorpora los comentarios de los clientes y requiere una intensa programación colaborativa da como resultado un software de alta calidad.

*“Optimiza la planificación y gestión de XP con una herramienta de gestión del trabajo que tenga la capacidad de actualizarse y adaptarse en tiempo real, tal como lo hace tu trabajo.”*

## **4. Conclusiones**

- Una comprensión detallada de los principios y valores de XP resalta su enfoque en la comunicación efectiva, la simplicidad, la retroalimentación continua, el coraje y el respeto. Estos valores forman la base de un enfoque que fomenta un entorno colaborativo y adaptativo.
- Cubre prácticas clave como planificación de juegos, lanzamientos pequeños, diseño simple, gestión de desarrollo de pruebas y más, proporcionando información práctica sobre el enfoque de XP para la programación. Planifique, cree y distribuya software.

Estas prácticas son consistentes con los valores y principios establecidos.

- Aunque esta investigación no proporciona estudios de casos específicos, su presentación de las etapas, principios y prácticas de XP muestra que el enfoque ha demostrado ser eficaz para los equipos. Requiere adaptabilidad y calidad en el desarrollo de software. Mencionar la intensidad de los procesos indica que un resultado positivo requiere un esfuerzo significativo.

## 5. Recomendaciones

- Es crucial que todos los miembros del equipo se comprometan con los valores fundamentales de la Programación Extrema (XP), como la comunicación efectiva, la sencillez en el diseño y el respeto en el trabajo en equipo. Este compromiso sienta las bases para una implementación exitosa.
- Proporcionar oportunidades de formación y capacitación continua para que los miembros del equipo se mantengan actualizados sobre las prácticas y principios de XP. Esto garantiza una comprensión profunda y la aplicación efectiva de la metodología.
- La Programación Extrema se basa en una comunicación abierta y directa. Fomentar un entorno colaborativo donde los miembros del equipo se sientan cómodos compartiendo ideas y retroalimentación contribuirá significativamente al éxito del proyecto.

## 6. Bibliografía/ Referencias

- SYDLE. (2021, febrero 18). *Extreme Programming: ¿qué es y cómo funciona?* Blog SYDLE; SYDLE. <https://www.sydle.com/es/blog/extreme-programming-602ee205da4d096809438c9c>
- *¿Qué Es Extreme Programming (XP)? - Valores, Principios Y Prácticas.* (2022, diciembre 22). Nimblework. <https://www.nimblework.com/es/agile/programacion-extrema-xp/>
- Raeburn, A. (2022, noviembre 28). *¿Qué es la programación extrema (XP)? [2022]* • Asana. <https://asana.com/es/resources/extreme-programming-xp>
- Canive, T. (2020, mayo 27). *Metodología XP o Programación Extrema.* Gestor de proyectos online. <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp>
- Mendoza, M. L. (2020, septiembre 18). *Extreme Programming: Qué es y cómo aplicarlo.*

*OpenWebinars.net.* <https://openwebinars.net/blog/extreme-programming-que-es-y-como-aplicarlo/>