



TECNOLÓGICO NACIONAL CEICOM  
CARRERA: SISTEMAS INFORMATICOS

**F**

Practica02

## **PROGRAMACIÓN EXTREMA**

Materia: Análisis y Diseño de Sistemas

Elaborado por: Fabrica Estevan Plaz

Docente: Ing. Baltazar Llusco Ever Jaime

Fecha: 14 de octubre de 2018

---

## ¿Qué es XP?

La programación extrema o XP (Extreme Programming) es una mitología ágil de desarrollo de software que intenta potenciar las relaciones entre los integrantes del proyecto y se concentra en la adaptabilidad. Según Kent Beck: “XP es un proceso ligero, de bajo riesgo, flexible, predecible, científico y divertido de desarrollo de software”. Las prácticas que generaron cierta desconfianza y rechazo cuando la metodología fue difundida son:

- Programación en pares
- Pruebas unitarias
- Pruebas continuas
- Refactorización

A diferencia de las metodologías tradicionales, XP está orientada al desarrollador, al usuario, al resultado real y al incremento de funcionalidad.

## Historia de XP

La programación extrema fue creada por Kent Beck, Ward Cunningham y Ron Jeffries en 1996, Beck llegó a la empresa y observó a los problemas del desarrollo estaban basados en las estructuras y las metodologías. Sus ideas originales sobre un buen desarrollo distaban demasiado de los modelos rígidos utilizados que hacían gran hincapié en el uso de extensa y refinada documentación. En 1999 Beck publicó el libro *Extreme Programming Explained*, que difundió y atrajo la atención sobre la metodología. Desde ese momento, XP se ha convertido en la principal metodología ágil de desarrollo, siendo atractivo para todo tipo de desarrollador.

## Cuando utilizar XP

La programación extrema fue concebida pensando en desarrollos que presentan las siguientes características:

- **Proyecto con requisitos variables, cambiantes:** casi todos los proyectos reales de software podrían entrar dentro de esta categoría
  - **Proyectos de alto riesgo:** son aquellos proyectos que proponen soluciones con calendarios realmente ajustados suponen un riesgo para su cumplimiento.
  - **Proyecto con pocos programadores:** las organizaciones con poco personal generalmente no pueden estructurarse de forma tal de dotar a todas las áreas de los profesionales adecuados. En esos casos, muchas deciden directamente no aplicar una metodología. XP permite equipos entre 2 y 12 desarrolladores puedan seguir perfectamente un proceso controlado, con normas de calidad y mejores prácticas sin necesidad de mayor cantidad de recursos humanos.
-

El tamaño del proyecto y el equipo han sido objeto de muchos estudios y pruebas de la metodología. En 2001, Michael Lauer presento una investigación sobre el uso de XP en proyectos propios en donde se trabajó con más de cuarenta desarrolladores, excediendo lo que se recomienda. Lauer propone lo siguiente:

- Adoptar XP cambiando la metáfora de sistema por un diseño confiable que deberá obtenerse al inicio.
- Contar con un equipo de arquitectos que definan y diseñen la arquitectura de componentes.
- Eliminar la programación en pares, la semana de 40 horas y el cliente en el sitio.
- Acortar todos los ciclos de desarrollo propuestos por XP.
- Capacitar continuamente al equipo de desarrolladores.
- Eliminar el mito de mes-hombre.

### **Promesas de la programación extrema**

Este tipo de metodologías hace un esfuerzo por promover la relación entre los miembros del equipo, dándoles un trabajo justo y prometiéndoles que obtendrán ciertas ventajas por su implementación.

#### **Promesas a los clientes y gestores**

- ❖ Obtención del máximo valor en cada semana de desarrollo.
- ❖ Progresos concretos.
- ❖ Modificación de requisitos sin asumir costos prohibitivos.

#### **Promesas a los desarrolladores**

- ❖ Trabajar en lo que realmente importa.
- ❖ Estarán acompañados por el equipo en las situaciones críticas.
- ❖ Mucho margen de acción en el desarrollo.
- ❖ Se enfocaran en el aspecto que les interesa del proyecto

### **Objetivos de la programación extrema**

El objetivo de la programación extrema está centrado en el resultado. Sin embargo, a diferencia de otras, en XP se hace foco en la satisfacción del cliente. Si intenta desarrollar el producto de acuerdo a las expectativas del usuario.

### **Valores de XP**

En las metodologías ágiles existen un conjunto de valores críticos y necesarios para que estas cumplan las expectativas. La programación extrema elige algunos valores y los promueve pensando, principalmente, en los aspectos humanos sobre las técnicas.

---

---

## **Simplicidad**

La base de la programación extrema es la simpleza. La simpleza propuesta por XP se refiere a evitar cualquier gasto innecesario, y en esto incluye la forma de desarrollar, las herramientas usadas y la cantidad de documentación necesaria. En el aspecto técnico se promueve el diseño simple, el cual garantiza agilizar los tiempos de desarrollo y mantenimiento.

Las ventajas de la simpleza son más evidentes a medida que crece el proyecto. La refactorización, el código compartido y la programación en pares son algunas de las prácticas para mantener el valor de la simpleza como esencia en XP.

## **Comunicación**

La comunicación es una parte vital del proceso. Esta metodología se sustenta como documentación del código. La comunicación sin barreras constante permite encontrar a la gente adecuada para el problema, evitando las inaccesibles cadenas de solicitudes.

## **Retroalimentación (feedback)**

Todo proyecto cambia constantemente y, por su puesto, cuando antes percibimos la variación podemos responder a ella. Se hace evidente al cliente en el sitio para poder interactuar con él, así como evitar barreras entre integrantes del equipo.

## **Coraje**

El coraje y la valentía son dos atributos necesarios en los directivos del proyecto.

Por ejemplo hacemos notar que uno de los valores que siempre se destacan es el respeto si esto no aplican los miembros del proyecto están destinados a fracasar en sus emprendimientos.

## **Prácticas de la programación extrema**

Ahora conoceremos sus doce prácticas, agrupadas en cuatro categorías.

### **Retroalimentación a escala fina**

Los procesos cambiantes suponen una flexibilidad y una adaptación al cambio sostenidas.

Retroalimentación del proceso y de su ejecución, nos permite ajustar en poco tiempo a aquellos elementos que nos pueden estar perjudicando, como también obtener beneficios. Muchas de las prácticas de XP tratan estos temas.

- ❖ El principio de pruebas: para alcanzar el objetivo se generan un plan de pruebas real que debe poder ser automatizado y permitir una visión realista sobre el estado del proyecto.
-

- ❖ **Proceso de planificación:** el cliente crea las historias de usuario, donde plasma sus requisitos. Las continuas reuniones entre el equipo y el cliente hacen que esta etapa sea productiva, ya que ellos se asignan las prioridades.
- ❖ **Cliente en el sitio:** el cliente trabaja en el lugar con el equipo de desarrollo, pudiendo interactuar directamente con los programadores. De esta manera se minimizan las barreras ocasionadas por la documentación.
- ❖ **Programación en pareja:** dos desarrolladores trabajan, codifican juntos en una sola estación de trabajo, reduciendo errores y ganando en tiempo de desarrollo. La programación en parejas es uno de los puntos con mayor resistencia.

### Proceso continuo en lugar de por lotes

En XP, en lugar de tener bloques de proceso que deben ser ejecutados con un estricto orden y con dependencias secuenciales, tenemos un conjunto de prácticas que se realizan a lo largo de la vida del ciclo de vida del producto.

- ❖ **Integración continua:** se generan versiones nuevas continuamente a lo largo del proceso de desarrollo.
- ❖ **Refactorización:** se realizan evaluaciones continuas del código y se reescribe la funcionalidad, mejorando el diseño.
- ❖ **Entregas pequeñas:** establecidas dentro de 4 semanas, donde se tiene una pequeña aplicación que funciona en vista real.

### Entendimiento compartido

Un equipo con conocimiento, predisposición al trabajo y enfocado en un objetivo produce más y mejor.

Compartir la información y el código posibilita que todos conozcan las tareas actuales y los esfuerzos necesarios para desarrollar.

- ❖ **Diseño simple:** es un nivel de sencillez aceptable que cumpla con los requisitos que ha efectuado el cliente.
- ❖ **Metáfora:** es la visión que los desarrolladores e integrantes del equipo del sistema completo. La metáfora puede ser mejorada e incrementada a medida que el desarrollo avanza.
- ❖ **Propiedad colectiva del código:** todos poseen acceso al código y esto permite que, al ser visualizado por más gente, posea menos errores.
- ❖ **Estándar de codificación:** de forma tal de producir código como si fuera escrito por una sola persona.

### Bienestar del programador

---

El programador como parte esencial del desarrollo de software debe trabajar en condiciones normales. La semana de 40 horas intenta llevar al programador a trabajar durante tiempos razonables para producir código de mayor calidad. XP nos presenta cuatro actividades básicas: codificar, hacer pruebas, escuchar y diseñar. Para todo el desarrollo XP se utilizan solo unos pocos artefactos:

- Historias del usuario
- Tareas de ingeniería
- Pruebas de aceptación
- Pruebas unitarias y de integración
- Plan de entrega
- Código

### **Actividades básicas**

Todo equipo de XP tiene, a lo largo del proceso, que cumplir con distintas responsabilidades. Las actividades básicas en XP son:

- **Escuchar:** la actividad está referida a escuchar al cliente y al equipo de desarrollo. Para llevar a cabo existen las reuniones.
  - **Diseñar:** obtener la información del usuario y trasladarla a documentos válidos para generar el código.
  - **Codificar:** escribir el código del producto, siguiendo los estándares y pautas establecidas.
  - **Probar:** realizar todas las actividades de prueba contempladas por el equipo.
-

