



# Instituto Politécnico Nacional

## Escuela Superior de Cómputo



# METODOLOGÍA Y PLANEACIÓN

Webelectivo

*Análisis y Diseño Orientado a Objetos*

Profesor: Chadwick Carreto Arellano

Grupo: 2CM7

Fecha: 28 / Febrero /2018

Alumnos:

- |                               |            |
|-------------------------------|------------|
| • Calva Hernández José Manuel | 2017630201 |
| • Ruíz López Luis Carlos      | 2014081397 |

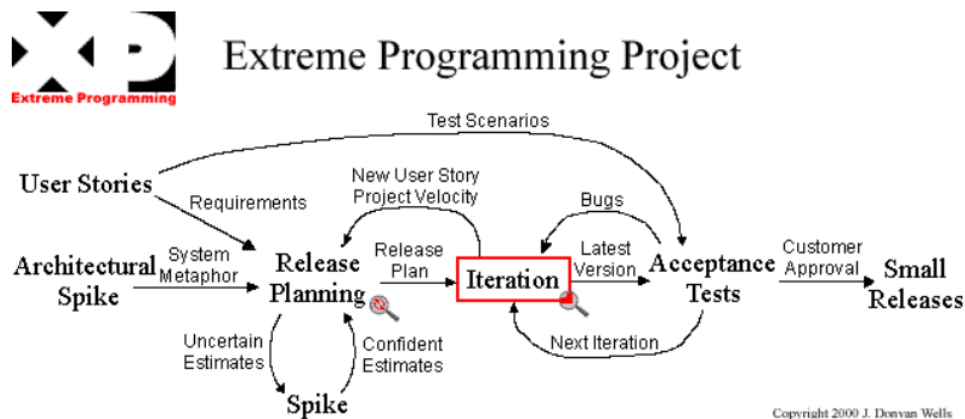
## Índice

Extreme Programming (XP).....	2
Justificación.....	2
Prácticas en XP .....	3
Cuando no es apropiado usar XP .....	4
Fases de la metodología .....	5
Planificación.....	7

## Extreme Programming (XP)

Extreme Programming es una disciplina de desarrollo de software que permite producir software de alta calidad, con un alto nivel de vida para el equipo de desarrollo. Está basada en valores como la simplicidad, comunicación, retroalimentación, coraje y respeto.

Trabaja por traer el equipo entero junto en la presencia de prácticas sencillas., con suficiente retroalimentación para permitir al equipo ver dónde están y perfeccionar el desarrollo para esa situación específica.



## Justificación

Se escogió la metodología XP (Extreme Programming) ya que se busca realizar un proyecto en relativamente poco tiempo y, a su vez, con un equipo de solamente dos personas. Dadas estas circunstancias, la metodología de XP nos permiten evitar tareas que pueden consumir mucho tiempo como lo es la documentación exhaustiva y proceder a un camino que se basa en la prueba / error.

Otro punto que nos llevó a elegir esta metodología es que no tenemos la certeza de cuáles serán la totalidad de las funcionalidades que debe incluir nuestro proyecto, ya que al no haber como tal un predecesor o algún otro sistema que nos sirva de modelo o referencia, estamos partiendo bajo ideas que posiblemente deban irse modificando o ajustando conforme se vayan generando las iteraciones.

## Prácticas en XP

1. **Proceso de planeación**

El cliente define las reglas de negocio de las características que necesita que sean terminadas y qué necesita puede ser diferido.

2. **Pequeños lanzamientos**

Se coloca un pequeño sistema en una producción temprana, al cual se le harán actualizaciones frecuentemente en ciclos cortos de tiempo.

3. **Metáfora**

Se utiliza un “sistema de nombres” común para todos, así como una descripción del sistema que permita guiar el desarrollo y la comunicación.

4. **Diseño simple**

Se busca cumplir de la forma más sencilla con los requerimientos, no se piensa a futuro.

5. **Pruebas**

XP se centra en verificar el sistema todo el tiempo, esto implica que durante el desarrollo se escriben las pruebas primero para que los requerimientos se vean reflejados en ellos. El cliente revisa las pruebas y verifica que las características solicitadas estén siendo cubiertas.

6. **Recodificación**

Durante todo el desarrollo se busca mejorar el diseño del sistema, esto se logra manteniendo el software limpio: sin duplicados, con una comunicación constante, completa y simple.

7. **Programación en pares**

El código se escribe en parejas en una misma computadora, ya que se considera que dos pares de ojos y dos cerebros pueden realizar una mejor codificación y detectar más errores.

8. **Propiedad colectiva**

Todo el código pertenece a todos los programadores, esto implica que cualquier persona puede modificar partes del código a su conveniencia, lo que permite un desarrollo más rápido.

9. **Integración continua**

Los equipos crean e integran el sistema de software múltiples veces por día, lo que permite que los demás estén actualizados respecto a los avances y puedan decidir su siguiente paso de manera más veloz.

10. **40 horas a la semana**

La metodología XP no busca explotar a los programadores, ya que al final resulta más ineficiente. Al trabajar el tiempo justo se busca gente más fresca, saludable y efectiva.

11. **Cercanía con el cliente**

El proyecto debe contar con una persona que responderá por el cliente respecto a determinar los requerimientos, establecer las prioridades y responder inmediatamente a los programadores.

12. **Codificación estándar**

Los programadores deben seguir el mismo camino en la codificación, de tal forma que las reglas que se establezcan permitan que la comunicación entre los miembros sea clara.

## Cuando no es apropiado usar XP

Desarrollo concurrente de middleware. El hecho de que existan múltiples escenarios, combinado con la imposibilidad de realizar pruebas unitarias confiables para un comportamiento simultáneo, hace que XP sea inaplicable. Algunas características de la metodología XP pueden ser aplicables, pero no es su totalidad.

Cuando se trata de núcleos del sistema operativo y controladores de dispositivos. Esto sigue un razonamiento similar al middleware concurrente.

En sistemas críticos para la seguridad donde el cambio debe ser manejado con mucho cuidado para preservar la seguridad. Recuerde: "Las pruebas sólo pueden mostrar la presencia de errores, no su ausencia".

Sistemas de 'Legado' donde el volumen de código supera con creces el tiempo disponible para mantenerlo, pero a veces es necesario 'retocar'. Desagradable, sí, pero a veces es necesario.

Cuando el proyecto entero está creando costosos cambios producto de las decisiones basadas en el software. Muchos de ellos probablemente cambien el "Sistema de Metáforas" también.

## Fases de la metodología

### 1. Fase de Exploración:

Se busca definir el alcance general del proyecto por medio las necesidades que establecerá el cliente mediante la redacción de sencillas “historias de usuarios”. Con base a ellas se estimarán tiempos de desarrollo; sin embargo, estos pueden variar cuando sean analizados a detalle, ya que de momento son dados con datos de muy alto nivel.

Esta fase dura típicamente un par de semanas y da como resultado una visión general del sistema con una aproximación superficial al estimado de tiempo general.

Durante la fase se realizan las siguientes acciones:

- El cliente establece las características que el software debe proveer.
- Los programadores dividen las características en tareas independientes y estiman el trabajo que necesitarán para completarlas.

### 2. Fase de planificación:

La planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y, asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas, o “Release Plan”.

Durante la fase se realizan las siguientes acciones:

- El cliente escoge las tareas más importantes que deben completarse para el siguiente lanzamiento.
- Los programadores escogen tareas y establecen parejas.

### 3. Fase de iteraciones:

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al

principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo.

Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance.

Durante la fase se realizan las siguientes acciones:

- Los programadores escriben pruebas unitarias.
- Los programadores añaden características que aprueben las pruebas unitarias.
- Los programadores arreglan problemas con las pruebas hasta que se aprueben todas ellas.
- Se integran los códigos.
- Se produce una versión de lanzamiento.
- El cliente corre pruebas para verificar las características.

#### **4. Fase de puesta en producción:**

Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa.

En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste (“fine tuning”).

Durante esta fase se realizan las siguientes acciones:

- La versión pasa a la fase de producción.
- Los programadores actualizan las proyecciones con base en la cantidad de trabajo que se concluyó en este ciclo de lanzamiento.
- Se vuelven a establecer prioridades respecto a las tareas que quedan por concluir y se repite el proceso.

# Webelectivo

Periodo resaltado: 60 Duración del plan Inicio real % Completado Real (fuera del plan) % Completado (fuera del plan)

ACTIVIDAD	ENCARGADO	INICIO DEL PLAN	DURACIÓN DEL PLAN	INICIO REAL	DURACIÓN REAL	PORCENTAJE COMPLETADO	PERIODOS																																	
							1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	
PROPUESTA DE PROYECTO	Equipo	1	1	1	2	100%	■	■																																
Problemática	José Manuel Calva Hernández	1	1	1	1	100%	■	■																																
Soluciones existentes	Equipo	1	1	1	1	100%	■	■																																
Propuesta de solución	José Manuel Calva Hernández	1	1	1	1	100%	■	■																																
PLANEACIÓN Y METODOLOGÍA	Equipo	2	1	2	2	100%	■	■	■																															
Metodología	Equipo	2	1	2	1	100%	■	■	■																															
Planeación	Luis Carlos Ruiz López	2	1	2	1	100%	■	■																																
VIABILIDAD Y ANÁLISIS DE REQUERIMIENTOS	Equipo	3	2	2	4		■	■	■	■																														
VIABILIDAD	Equipo	3	2	1	4	0%	■	■	■	■	■																													
DEFINICIÓN DEL PROCESO	José Manuel Calva Hernández	3	1	1	5	50%	■	■	■	■	■																													
REQUERIMIENTOS FUNCIONALES	Equipo	4	1	3	1	0%			■	■	■	■																												
REQUERIMIENTOS NO FUNCIONALES	Equipo	4	1	3	1	0%			■	■	■	■																												
EXPLORACIÓN	Equipo	4	4	1	9	0%	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
Definición de Historias de Usuario	Equipo	4	1	1	8	0%	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
Pruebas de solución	Equipo	5	1	5	2	0%					■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Establecer tareas independientes	Equipo	6	2	6	3	0%						■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Estimacion de tiempos de desarrollo	José Manuel Calva Hernández	7	1	7	2	0%							■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
PLANIFICACIÓN	Equipo	8	2	7	4	0%								■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Realización de reunión grupal	Equipo	8	2	8	2	0%									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Establecer la Metáfora del Sistema	Equipo	8	2	8	2	0%										■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Definir requerimientos	Equipo	8	2	8	2	0%											■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Acordar implementación de Historias de Usuario	Equipo	9	1	8	2	0%												■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Crear tarjetas CRC	Equipo	9	1	8	2	0%													■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Establecere prioridades	Equipo	9	1	8	2	0%														■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Determinar el Plan de Lanzamiento	Equipo	9	1	8	2	0%															■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
ITERACIONES	Equipo	10	4	9	6	0%																■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
Escribir pruebas unitarias	Equipo	10	1	10	1	0%																	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Añadir características a las pruebas unitarias	Equipo	10	1	10	3	0%																		■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Arreglar problemas con pruebas unitarias	Equipo	11	1	10	2	0%																			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Integracion de los codigos	Equipo	11	1	11	2	0%																				■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Producir version de lanzamiento	Equipo	12	2	12	2	0%																					■	■	■	■	■	■	■	■	■	■	■	■	■	■
Verificación de la versión de lanzamiento por parte del cliente	Equipo	13	1	13	2	0%																						■	■	■	■	■	■	■	■	■	■	■	■	■
PUESTA EN PRODUCCIÓN	Equipo	14	2	13	4	0%																							■	■	■	■	■	■	■	■	■	■	■	■
Aprobación del cliente	Equipo	14	1	13	2	0%																								■	■	■	■	■	■	■	■	■	■	■
Versión de lanzamiento ingresa a fase de producción	Equipo	14	1	12	3	0%																									■	■	■	■	■	■	■	■	■	■
Revision del Plan de entregas	Equipo	15	1	15	1	0%																										■	■	■	■	■	■	■	■	■
Actualización de proyecciones	Equipo	15	1	15	2	0%																											■	■	■	■	■	■	■	■
ITERACIONES	Equipo	16	4	15	6	0%																											■	■	■	■	■	■	■	■
Escribir pruebas unitarias	Equipo	16	1	16	1	0%																												■	■	■	■	■	■	■
Añadir características a las pruebas unitarias	Equipo	16	1	16	3	0%																													■	■	■	■	■	■
Arreglar problemas con pruebas unitarias	Equipo	17	1	16	2	0%																														■	■	■	■	■
Integracion de los codigos	Equipo	17	1	17	2	0%																															■	■	■	■



