

# Dividir Tarea



English

## Entradas

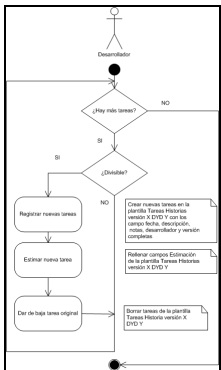
- Tareas Historias versión X
- Tareas Historias versión X Programador Y

## Salidas

- Tareas Historias versión X
- Tareas Historias versión X Programador Y

## Solución

## Proceso



## Tiempo de Desarrollo

- ◆ Para adquirir el conocimiento necesario para desarrollar el producto software:
- ◆ Para crear el Patrón de Producto: 45 minutos.
- ◆ Para aplicar el Patrón de Producto:

## Video Explicación

- No aplica

## Patrones Relacionados

- Patrón Estimar Tarea

## Controladores de Calidad

- Ninguno

## Plantillas

Tareas\_Historias\_Ver\_X.doc  
Tareas\_Historias\_Ver\_X\_DyD\_Y.doc

## Ejemplos

- Ninguno

## Herramientas de Soporte

- Será necesario un editor de texto como [OpenOffice Writer](#) o [Microsoft Word](#).
- Además de una herramienta como [Visual Paradigm for UML](#) para la realización de los diagramas expuestos.



## Contexto Inicial

Este producto puede usarse en cualquier proyecto en el que debido al resultado de la estimación de una tarea, por el cual no pueda completarse en tiempo, esta deba dividirse en varias más sencillas para permitir su finalización a tiempo.



## Contexto Resultante

Los desarrolladores obtienen nuevas tareas como resultado de la separación de otras más costosas en tiempo.



## Problema

Los desarrolladores deben ser capaces de estimar cada tarea, y de dividir las tareas que necesiten más de unos días en ser implementadas o bien agruparlas.



## Restricciones (*Forces*)

- **Características de las organizaciones:** Este patrón puede utilizarse en los proyectos existentes en cualquier tipo de compañía.
- **Tipo de sistema a desarrollar:** Este producto puede utilizarse en proyectos en los que los requerimientos de usuario sean cambiantes.
- **Tipo de Cliente:** Debe existir, o debe conseguirse, que el área de negocio destinataria del desarrollo se implique en la consecución del mismo.
- **Heurísticas de uso:** Si se necesita disponer urgentemente del aplicativo o de algunas de sus funcionalidades. Edit section



## Roles

- Desarrolladores (2 a 12)



## Lecciones Aprendidas

- El programador debe dividir una tarea cuya estimación supere un cierto número de días.
- Los desarrolladores dividen las tareas si la estimación de alguna de ellas supera unos pocos días.
- En el documento de salida Tareas Historias versión X se añaden al final del documento las nuevas tareas resultantes de la división de las que tenían el campo en cuestión marcado, eliminando estas últimas tareas.
- En el documento de salida Tareas Historias versión X Programador Y se añaden al final del documento las nuevas tareas resultantes de la división, eliminando estas últimas.



## Nivel de Madurez

- Este Patrón de Producto no se relaciona con ningún nivel de madurez(N/A).



## Conocimientos y Habilidades Básicos



### Conocimientos

- Conocimiento del estándar de codificación que define la propiedad del código compartido así como las reglas para escribir y documentar el código y la comunicación entre diferentes piezas de código desarrolladas por diferentes equipos. Los programadores las han de seguir de tal manera que el código en el sistema se vea como si hubiera estado escrito por una sola persona.
- Conocimiento de la visión común de cómo funciona el programa en el que se desarrollan las actividades.



### Habilidades

- Capacidad de trabajo en grupo. Todos en un equipo XP contribuyen de la manera que pueden.
- Predicción de qué se habrá terminado para la fecha de entrega, y determinación de qué hacer después.
- Capacidad de programación de a pares. Además de generar mejor código y pruebas, sirve para comunicar el conocimiento a través de los equipos.



## Recursos de Información

- Álvarez, José R. y Arias Manuel. Método Extreme programming. Recuperado el 2010-03-05 de <http://www.ia.uned.es/ia/asignaturas/adms/GuiaDidADMS/node61.html>
- Anaya Villegas, Adrian. A propósito de programación extrema XP (extreme Programming). Recuperado el 2010-02-10 de <http://www.monografias.com>
- Beck, K. (2000), Una explicación de la programación extrema. Aceptar el cambio. Ed. Addison Wesley.
- De Seta, Leonardo. Una introducción a Extreme Programming. Recuperado el 2010-03-02 de <http://www.dosideas.com/noticias/metodologias/822-una-introduccion-a-extreme-programming.html>
- Extreme Programming: A gentle introduction. Recuperado el 2010-03-15 de <http://www.extremeprogramming.org/>
- Joskowicz, José. Reglas y prácticas en Xtreme Programming. Recuperado el 2010-03-15 de <http://ie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- Letelier, Patricio y Panadés M<sup>a</sup> Carmen. Metodologías Ágiles en el desarrollo de software: extreme programming. Recuperado el 2010-03-15 de <http://www.willydev.net/descargas/masyxp.pdf>
- Newkirk, James y Martin, Robert C. (2001), La programación Extrema en la Práctica. Ed Addison Wesley.