

# Generar Código



English



## Entradas

- Diseño Tarea



## Salidas

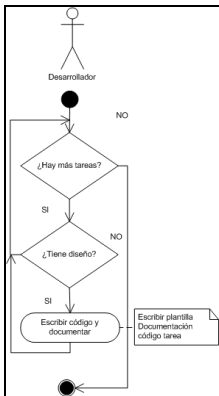
- Código Tarea
- Documentación Código Tarea



## 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 Diseñar
- Patrón Registrar Progreso



## Controladores de Calidad

- Ninguno



## Plantillas

- Código\_Tarea.xx
- Documentación\_Código\_Tarea.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 se siga la filosofía de trabajo en parejas para la implementación de código.

En XP todo el software productivo se escribe en pareja, dos programadores sentados lado a lado en una misma computadora.

Esta práctica asegura que todo el código productivo fue revisado por al menos otro programador, y genera mejores diseños, mejores pruebas y mejor código. Puede parecer ineficiente que dos programadores hagan el "trabajo de un programador", pero lo contrario es cierto. Los estudios sobre la programación de a pares muestran que las parejas producen mejor código en aproximadamente el mismo tiempo que un programador trabajando solo.



## Contexto Resultante

Se obtienen pequeñas piezas de código implementadas por dos personas, lo cual reduce el porcentaje de errores (principio de los cuatro ojos).

Los equipos XP usan un estándar de código en común, de manera que el código del sistema se vea como si fuera escrito por una única persona muy competente. No importa mucho el estándar en si mismo: lo importante es que el código se vea familiar, para permitir la propiedad colectiva.



## Problema

La programación en parejas no consiste en que una persona escriba y la otra mire, es un dialogo entre dos personas que intentan simultáneamente programar, diseñar, analizar y probar.



## 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.



## Roles

- Desarrolladores (2 a 12)



## Lecciones Aprendidas

- La programación en parejas no es una tutoría. Si en la pareja hay un programador experto y otro inexperto, el primero debe asesorar y guiar al otro para que al cabo de un corto tiempo alcance el nivel requerido.

Las parejas de desarrolladores deben estar bien organizadas. Si se detecta que la relación entre los miembros de una pareja no es buena, conviene perder un cierto tiempo en la reorganización de las mismas. Lo mas importante para que el trabajo en equipo funcione es la comunicación entre los miembros de la pareja de programadores. El código implementado debe ser sencillo, para lo cual se deben seguir las recomendaciones:

1. El sistema (código y pruebas) debe comunicar todo lo que se quiere comunicar y nada mas.
2. El sistema no debe tener código duplicado
3. El sistema debería obtener el menor numero posible de clases
4. El sistema debería obtener el menor numero posible de métodos

Debemos realimentarnos de lo aprendido en el diseño. Se debe hacer de forma cíclica todo el proceso tan pronto como sea posible:

1. Inversión inicial pequeña
  2. Asumir la simplicidad
  3. Cambio incremental
  4. Viajar ligero (no intentar cubrir expectativas que no esta descritas en el diseño)
- Conviene escribir código de manera conjunta entre dos personas que colaboran entre sí.
  - El archivo Diseño Tarea es un documento con el diseño de la tarea a implementar.
  - El producto de salida Código fuente es el código fuente que implementa la tarea en cuestión.
  - El documento Documentación Código Fuente es la documentación que explica el código implementado.



## 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 proposito 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://iie.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.
-