

# Integrar



English

## Entradas

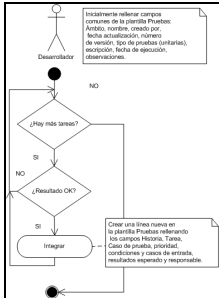
- Pruebas

## Salidas

- Pruebas

## 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 Pruebas Unitarias
- Patrón Modificar Casos de Prueba
- Patrón Pruebas de Integración

## Controladores de Calidad

- Ninguno

## Plantillas

- Pruebas.xls

## 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 los que tras el diseño u la implementación de código o modificación del mismo deba integrarse con el resto de trabajo realizado por el equipo.



## Contexto Resultante

El trabajo realizado por un miembro del equipo funciona 100% con el resto de trabajos realizado por todo el equipo.



## Problema

Los desarrolladores deben mantener equilibrado e integrado su trabajo con el del resto del equipo. La integración continua implica que debe integrarse el trabajo individual cada pocas horas (1 día como máximo).

En la integración se juntan los pequeños módulos de software se tienen y que parece que funcionan, pero resulta que esos pequeños trozos solo han superado pruebas de unidad donde no han tenido que interactuar con otros módulos. El problema es que existen errores que solo surgen en esa interacción y ese es precisamente el problema de la integración.



## 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

- Es importante tener una herramienta de integración que soporte un ciclo rápido de integración/construcción/pruebas. El conjunto de pruebas deben ejecutarse en unos pocos minutos. Además, las colisiones deben requerir un esfuerzo relativamente pequeño. Se reduce de manera drástica el riesgo del proyecto. Si dos personas tienen diferentes ideas sobre la forma o el funcionamiento de una parte del código se sabrá en horas. Nunca se dedicarán muchas horas a perseguir un error que fue cometido en algún momento de las últimas semanas.
- Hay que integrar cada vez que se desarrolle una clase o método para mantener el equilibrio y la unidad de todo el trabajo conjunto del equipo.
- El archivo Pruebas es un documento genérico, para todos los tipos de prueba con la casilla pruebas unitarias marcada, que aglutina todos los casos de prueba para una tarea en cuestión. Hay que poner especial atención en aquellos casos de prueba no satisfactorios, pues han de ser modificados. En este documento se refleja su responsable e historia asociada. Se pueden añadir o eliminar casos de prueba según se considere oportuno.



## 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.
-