

# Registrar Progreso



English



## Entradas

- Tareas Historias versión X



## Salidas

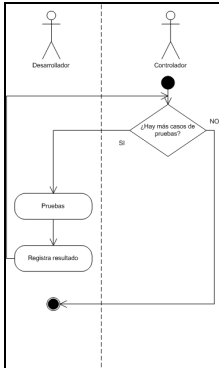
- Tareas Historias versión X (Indicadores de Progreso / Informe de Seguimiento)



## 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 Generar Código
- Patrón Recuperación Tareas
- Patrón Escribir Pruebas Funcionales para Historia



## Controladores de Calidad

- Ninguno



## Plantillas

- Tareas\_Historias\_Ver\_X.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 deba registrar el progreso de los programadores en la implementación de las tareas que les fueron asignadas.



## Contexto Resultante

El controlador obtiene una visión clara del estado de desarrollo de la versión en cuestión.



## Problema

El controlador, en función del balanceo establecido por cada programador, y el resultado reflejado en el informe de situación con el registro del progreso, debe evaluar la situación y recomendar la toma de decisiones para asegurar la consecución del proyecto en los plazos fijados.



## 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

- Controlador (1)



## Lecciones Aprendidas

- El controlador debe establecer conclusiones a raíz de la recogida de información para poder ofrecer al director del proyecto un estado del desarrollo del mismo. Se deben realizar informes de seguimiento periódicamente para tener el control sobre el desarrollo del mismo.
- El controlador pregunta a cada programador cuanto tiempo ha dedicado a cada una de sus tareas y cuanto le queda para completarlas.
- El documento Tareas Historias versión X contiene las tareas del ámbito técnico correspondientes a las historias de la versión en cuestión. A la salida, el campo responsable de realización de cada tarea ha de estar relleno, así como los campos ¿tiempo empleado en tarea? y ¿tiempo que resta para completarla?.



## 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.
- Se debe tener un conocimiento global del sistema y del trabajo del equipo para realizar un buen informe del estado de desarrollo.



### 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%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.