

## ***White\_Watch:***

### **Un Método para el Desarrollo de Proyectos Pequeños de Software**

Versión 2.0 –2013

Judith Barrios A. y Jonás Montilva C.

Universidad de Los Andes, Facultad de Ingeniería, Escuela de Ingeniería de Sistemas,  
Departamento de Computación, Grupo GIDyC, Mérida, Venezuela

{ijudith, jonas@ula.ve}

#### **1. Introducción**

El método ***White\_Watch*** es la versión *liviana* del método *Watch*. *W\_Watch* es un marco metodológico que describe, el conjunto estructurado de actividades necesarias para desarrollar un producto de software pequeño, de baja complejidad y con documentación técnica precisa.

En esta versión se trata de disminuir la elaboración detallada de documentos y/o especificaciones de apoyo parcial al proceso de desarrollo, permitiendo, al equipo de desarrollo pequeño (1 o 2 personas), dedicar más tiempo a las actividades de implementación e implantación de versiones operativas y evolutivas del producto. El rol de líder de proyecto es ejecutado en paralelo, y sin sobrecarga, por uno de los miembros del equipo mientras éste ejecuta otros roles técnicos. Es por ello que, las actividades gerenciales de control de calidad y de configuración, las cuales son indispensables en todo proyecto de software, se limitan a prescribir las actividades básicas de control de cambios, de validación y de verificación de especificaciones técnicas y de productos parciales y finales.

#### **2. Modelo de Procesos del Método**

El modelo de procesos esta organizado en dos grupos de procesos complementarios:

- los procesos gerenciales que incluyen los procesos de soporte, y
- los procesos técnicos de desarrollo del producto de software

El modelo de procesos del método ***W\_Watch*** inspirado en la metáfora del reloj de pulsera (*Watch* en Inglés), organiza los procesos técnicos, en forma circular, en las posiciones del dial de un reloj y ubica los procesos gerenciales en el centro – motor de control, de manera que éstos puedan controlar la ejecución de los procesos técnicos.

Esta manera de estructurar el marco metodológico permite que la ejecución de los procesos de desarrollo sea **cíclica, iterativa y controlada**. Los procesos gerenciales determinan, supervisan y controlan los ciclos, las iteraciones y la entrega de los productos parciales y finales del proyecto.

**Cada ciclo** de procesos **técnicos** produce una nueva **versión** del sistema (**enfoque progresivo o evolutivo**) o un nuevo **subsistema** del sistema **en desarrollo**, si se ha seleccionado el enfoque de **desarrollo incremental**.

En **cada ciclo** se puede **iterar entre los procesos técnicos** a fin de corregir errores, introducir nuevos requisitos o, simplemente, mejorar el producto en desarrollo. Son los procesos gerenciales quienes determinan el alcance de cada iteración y cuántas iteraciones se realizan por ciclo.

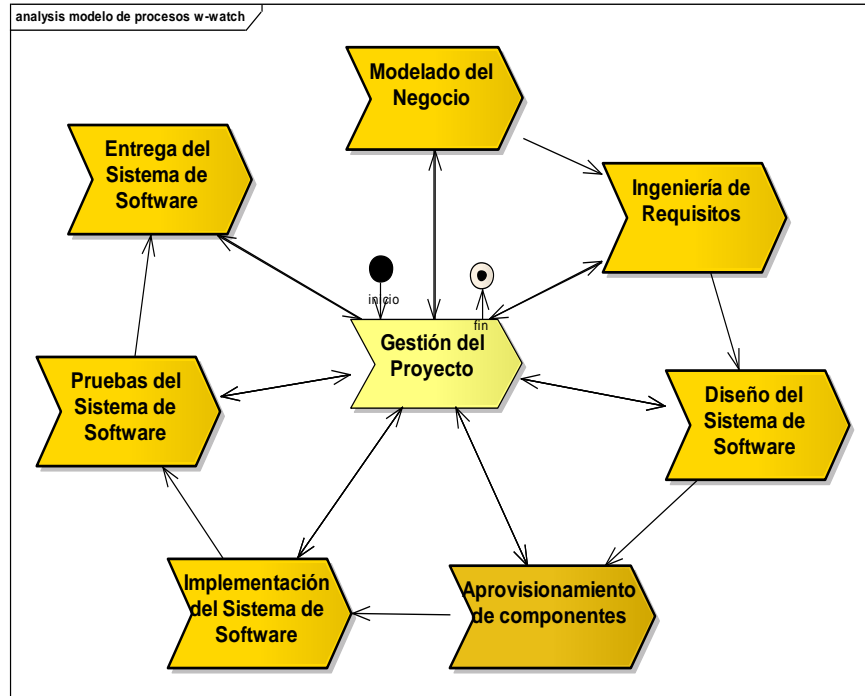


Figura 1. Modelo de Procesos del Método W\_Watch

Los **procesos gerenciales** describen las actividades que el líder del proyecto debe realizar para:

- Planificar, organizar y controlar el proceso de desarrollo del proyecto
- Asegurar la calidad del sistema mediante validaciones y verificaciones
- Gestionar los cambios en las especificaciones del producto

Los **procesos técnicos** son los procesos que prescriben lo que debe hacer el grupo de desarrollo para elaborar un producto de software pequeño y poco complejo. Contemplan actividades relacionadas con el Modelado de Negocios o dominio, la Ingeniería de requisitos, el Diseño de software, el Aprovisionamiento de Componentes, la Implementación, las Pruebas y la Entrega de producto parcial o final.

Se establecen un conjunto de **supuestos** que soportan el conjunto de actividades prescritas en el modelo de procesos de la figura 1. Estos son:

- 1) El personal de desarrollo en sus diferentes roles: líder de proyecto, analista, diseñador y programador, cuenta con las habilidades, experiencia y conocimientos relacionados con el uso de los lenguajes y de las herramientas de apoyo que se van a utilizar.
- 2) El enfoque de desarrollo evolutivo o incremental es la base para la planificación del número de iteraciones y ciclos que se realizarán.
  - a. Se parte del principio que la primera versión del producto es ya operativa y que cada nueva versión es el resultado de un refinamiento de la versión previa.
  - b. Como refinamiento se puede entender: corrección de deficiencias y de errores de la versión previa (o incremento), inclusión de nuevos requisitos (técnicos o del usuario) y/o mejoramiento del desempeño del producto.

- 3) El proceso de desarrollo se fundamenta en la máxima reutilización de componentes o módulos de software como medio para acortar el tiempo de entrega de versiones parciales de producto y por consiguiente, el tiempo total de desarrollo del software.
- 4) Se espera que se empleen herramientas de apoyo automatizadas, basadas en la notación UML, para la elaboración y mantenimiento de la documentación técnica del proyecto. Estas herramientas deberían facilitar las actividades de diagramación, especificación, modificación, traza y seguimiento de cambios y/o modificaciones en las especificaciones técnicas del producto elaborado.

Cada uno de los procesos del modelo de procesos se estructura en pasos y actividades, buscando disminuir la complejidad e indicar el cómo se realizan. Cada proceso es descrito mediante una tabla que asocia pasos (subprocesos) y actividades que indican de manera detallada el conjunto de acciones a ejecutar para llevar el proceso prescrito, y por consiguiente, cada paso del proceso. Ver figura 2.

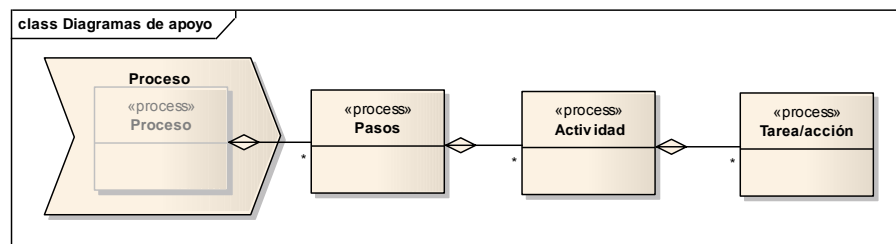


Figura 2. Estructura del Modelo de Procesos del Método *W\_Watch*

Como toda **guía metodológica** el modelo de procesos del *W\_Watch* debe ser **adaptado**, por el líder del proyecto, según las particularidades de cada proyecto de desarrollo.

Entre los **factores** a considerar para la adaptación se tienen: las **características** propias de cada **producto** y de los **ambientes** de desarrollo y de operación; se consideran además, los **recursos** utilizables tanto a nivel de **personal** como de **HW** y **SW** y las habilidades y destrezas requeridas por los miembros que conformarán el equipo de desarrollo.

En la figura 3 se muestra el diagrama de flujo de trabajo del Modelo de Procesos del método *W\_Watch*.

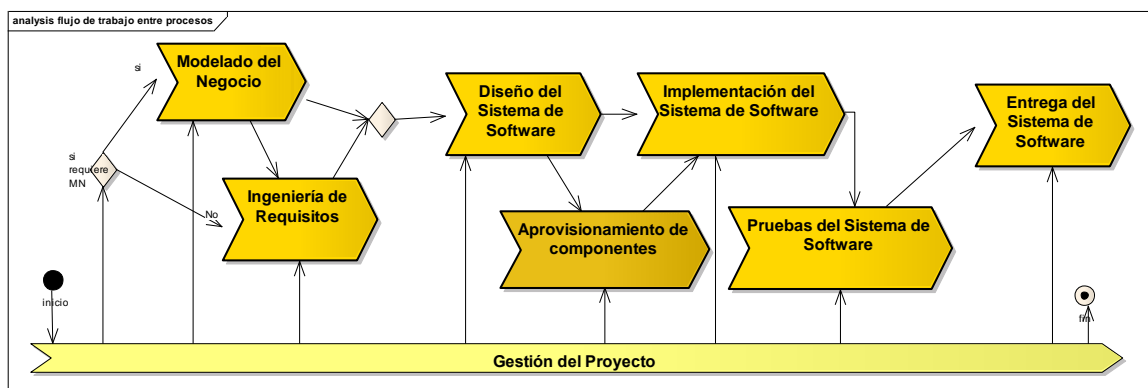


Figura 3. Flujo de trabajo del Modelo de Procesos del Método *W\_Watch*

Por ejemplo, considerando el modelo de la figura 3, en los proyectos que no desarrollan aplicaciones empresariales o cuyo producto de SW es empotrado, no se requiere realizar el

modelado del sistema de negocios; así, el proceso de desarrollo se iniciaría directamente con el proceso de Ingeniería de Requisitos.

A continuación se define el conjunto de pasos y actividades para cada uno de los procesos del modelo presentado en la figura 3.

### **3. Procesos de desarrollo W Watch**

#### **3.1. Proceso Gerencial: Gestión del Proyecto**

Pasos	Actividades	Técnicas y Notaciones	Productos
<ul style="list-style-type: none"> <li>Planificación del Proyecto</li> <li>Organización del grupo de desarrollo</li> <li>Control del proyecto</li> <li>Verificación y Validación (V&amp;V)</li> <li>Resolución de Riesgos</li> <li>Gestionar cambios en los requisitos del SW</li> <li>Control de Documentación</li> <li>Control de la configuración del software</li> </ul>	<ul style="list-style-type: none"> <li>Establecer la visión del producto</li> <li>Establecer disponibilidad de recursos</li> <li>Elaborar plan del proyecto</li> <li>Definir equipo de desarrollo y responsabilidades</li> <li>Definir lista de riesgos y contingencias</li> <li>Organizar reuniones de avance del proyecto</li> <li>Actualizar el plan</li> <li>Realizar las revisiones técnicas</li> <li>Analizar viabilidad de cambios en requisitos</li> <li>Actualizar documentación técnica</li> <li>Elaborar documentos de la aplicación</li> </ul>	<ul style="list-style-type: none"> <li>PERT/CPM</li> <li>Estructuras de grupos</li> <li>Estimación de costos</li> <li>Técnicas de V &amp; V</li> <li>Técnicas de gestión de riesgos</li> <li>Inspección de diseño y código</li> <li>Recorridos estructurados</li> <li>Técnicas de elaboración de documentos técnicos</li> <li>Matrices y listas de rastreo de requisitos</li> <li>Técnicas de gestión de configuración de software -SCM</li> </ul>	<ul style="list-style-type: none"> <li>Visión del producto</li> <li>Plan del Proyecto</li> <li>Lista de chequeo de riesgos</li> <li>Informe de V &amp; V</li> <li>Documentos del proyecto – informes</li> <li>Documentos de la aplicación</li> <li>Especificaciones actualizadas</li> </ul>

### 3.2. Procesos de Desarrollo - Proceso: Modelado de Negocios

Pasos	Actividades	Técnicas y Notaciones	Productos
1. Modelado de Objetivos del Sistema de Negocios - SN- contexto o dominio de la aplicación	<ul style="list-style-type: none"> <li>Establecer el alcance del sistema de negocios</li> <li>Definir objetivos de SN</li> <li>Elaborar una jerarquía de objetivos del SN (si necesario)</li> <li>Identificar los subsistemas del SN (si requerido)</li> </ul>	<ul style="list-style-type: none"> <li>Revisión de los manuales de organización</li> <li>Entrevistas con los involucrados en el SN</li> <li>Revisión de los manuales de organización</li> <li>Entrevistas con los involucrados en el SN</li> <li>Modelado de objetivos</li> </ul>	<ul style="list-style-type: none"> <li>Documento de definición del SN y de su alcance</li> <li>Diagrama de Objetivos del SN</li> </ul>
2. Modelado de los Procesos de Negocio del SN	<ul style="list-style-type: none"> <li>Modelar la cadena de valor del SN (si necesario)</li> <li>Modelar los procesos vitales o fundamentales del SN</li> <li>Modelar los procesos de soporte (o de apoyo)</li> <li>Modelar las actividades de los procesos de la cadena de valor asociados con el sistema de software a desarrollar</li> </ul>	<ul style="list-style-type: none"> <li>Observación y Entrevista con los expertos del SN</li> <li>Revisión de documentación técnica</li> <li>Modelado de Cadenas de Valor</li> <li>Modelado de Procesos en UML <i>Business</i></li> <li>Modelado de Actividades en UML</li> </ul>	<ul style="list-style-type: none"> <li>Cadena de Valor del SN</li> <li>Diagramas de Procesos del SN: <ul style="list-style-type: none"> <li>Jerarquía de procesos</li> <li>Descripción de procesos</li> </ul> </li> <li>Diagramas de actividades: <ul style="list-style-type: none"> <li>Diagramas de actividades x proceso</li> </ul> </li> </ul>
3. Modelado de actores/unidades organizacionales	<ul style="list-style-type: none"> <li>Identificar actores/unidades (pueden ser otros sistemas)</li> <li>Definir roles de los actores en cada proceso detallado en diagramas de actividades</li> <li>Completar diagramas de actividades (si necesario)</li> <li>Elaborar la matriz actores/procesos (si necesario)</li> </ul>	<ul style="list-style-type: none"> <li>Modelado de actividades con actores</li> </ul>	<ul style="list-style-type: none"> <li>Diagrama de Actividades por actor</li> <li>Descripción de Actores y sus Roles</li> <li>Diagrama de actividades – colaboración- entre actores - <i>swimlines</i></li> <li>Matriz Actores-Procesos</li> </ul>
4. Modelado de los Objetos de Negocio del SN	<ul style="list-style-type: none"> <li>Identificar los objetos de negocio y sus tipos para los procesos detallados y asociados con el sistema de software a desarrollar</li> <li>Definir las relaciones entre tipos de objetos</li> <li>Elaborar el modelo preliminar de objetos</li> </ul>	<ul style="list-style-type: none"> <li>Modelado de Clases en UML - representación de los conceptos manejados por el negocio</li> </ul>	<ul style="list-style-type: none"> <li>Diagramas de clases en UML</li> </ul>
5. Identificación de las Reglas de Negocio (si requerido)	<ul style="list-style-type: none"> <li>Identificar las reglas de negocio implicadas en los procesos del sistema de software a desarrollar</li> <li>Analizar y clasificar las reglas de negocio (alto y bajo nivel) si necesario</li> <li>Describir las reglas de negocios de bajo nivel si necesario)</li> </ul>	<ul style="list-style-type: none"> <li>Consultas a usuarios y expertos</li> <li>Búsqueda de documentación</li> <li>Modelado de reglas de negocio</li> </ul>	<ul style="list-style-type: none"> <li>Lista reglas de negocio del SN <ul style="list-style-type: none"> <li>Lista de las reglas de Alto nivel</li> <li>Descripción de las reglas de Bajo nivel</li> </ul> </li> </ul>
6. Modelado de Eventos (si requerido)	<ul style="list-style-type: none"> <li>Identificar eventos</li> <li>Asociar eventos con procesos</li> <li>Modelar el flujo de trabajo asociado a cada evento</li> <li>Elaborar la matriz de eventos-procesos</li> </ul>	<ul style="list-style-type: none"> <li>Modelado de eventos en UML <i>Business</i></li> </ul>	<ul style="list-style-type: none"> <li>Diagrama de eventos en UML <i>Business</i></li> <li>Diagramas de flujo entre procesos</li> <li>Matriz Eventos-Procesos</li> </ul>
7. Integrar los modelos	<ul style="list-style-type: none"> <li>Verificar coherencia entre los modelos elaborados</li> <li>Ensamblar el documento de modelado</li> </ul>	<ul style="list-style-type: none"> <li>Matriz de relación objetos/ procesos</li> <li>Técnicas de documentación</li> </ul>	<ul style="list-style-type: none"> <li>Modelo de Negocios del sistema de negocios - MN <ul style="list-style-type: none"> <li>Todos los modelos integrados</li> </ul> </li> </ul>

### 3.3. Procesos de Desarrollo – Proceso: Ingeniería de Requisitos

Actividades	Tareas	Técnicas y Notaciones	Productos
1.Descubrimiento de Requisitos	<ul style="list-style-type: none"> <li>Identificar y analizar los problemas de información que tiene el Sistema de Negocios (o el contexto - dominio)</li> <li>Determinar los objetivos del sistema de SW</li> <li>Identificar y clasificar los involucrados (<i>stakeholders</i>) y usuarios (internos y externos) – si MN a partir de modelo de actores</li> <li>Recolectar los requisitos que tienen los involucrados y usuarios - si hay un MN hacerlo a partir de modelo de actividades/actores</li> <li>Identificar requisitos de información a partir de los diagramas de procesos y de actividades (o del contexto)</li> </ul>	<ul style="list-style-type: none"> <li>Entrevista</li> <li>Documentación relacionada con el dominio – o el MN</li> <li>Observación de las actividades que realizan los usuarios</li> <li>Plantilla de definición de requisitos Volère [VOL04]</li> <li>Reuniones con usuarios</li> </ul>	<ul style="list-style-type: none"> <li>Listado de requisitos del cliente C</li> <li>Listado de requisitos técnicos</li> <li>Planillas Volère de requisitos</li> </ul>
2.Análisis de Requisitos	<ul style="list-style-type: none"> <li>Clasificar los requisitos F y NF</li> <li>Chequear requisitos <ul style="list-style-type: none"> <li>Comprobar necesidad, prioridad, consistencia, completitud y factibilidad</li> </ul> </li> <li>Negociar requisitos <ul style="list-style-type: none"> <li>Discutir, priorizar y acordar requisitos con el cliente y los usuarios de la aplicación</li> </ul> </li> <li>Elaborar cuadro detallado de los requisitos clasificados indicando sus prioridades y su fuente</li> <li>Validar requisitos con el cliente y usuarios</li> </ul>	<ul style="list-style-type: none"> <li>Técnicas de negociación</li> <li>Matriz de interacción entre requisitos</li> </ul>	<ul style="list-style-type: none"> <li>Documento de Definición de Requisitos (DDR) validado <i>Incluye Matriz de interacción entre requisitos</i></li> </ul>
3.Especificación de Requisitos	<ul style="list-style-type: none"> <li>Elaborar los diagramas de casos de uso</li> <li>Elaborar un diagrama preliminar de clases de objetos para cada caso de uso <ul style="list-style-type: none"> <li>Completar el modelo de objetos del negocio si MN</li> <li>Integrar los diagramas de clases</li> </ul> </li> <li>Elaborar diagramas de transición de estados (<i>si requerido</i>)</li> <li>Elaborar diagramas de secuencia entre clases de objetos del negocio – (<i>si requerido</i>)</li> <li>Integrar diagramas en documento de Especificación de Requisitos (DER)</li> <li>Realizar la revisión técnica del DER con el cliente, usuarios especializados y diseñadores</li> </ul>	<ul style="list-style-type: none"> <li>Refinamiento de modelo de objetos del negocio (MN)</li> <li>Modelado de sistemas en UML: <ul style="list-style-type: none"> <li>Diagramas de casos de uso</li> <li>Diagramas de clases</li> <li>Diagramas de estado</li> <li>Diagramas de secuencias</li> </ul> </li> </ul>	<p>Documento de Especificación de Requisitos (DER) validado <i>Incluye los modelos de especificación de requisitos</i></p>

### 3.4. Procesos de Desarrollo - Proceso: Diseño de software

Actividades	Tareas	Técnicas y Notaciones	Productos
1. Definición de la estructura inicial de la aplicación	<ul style="list-style-type: none"> <li>Determinar requisitos a implementar a partir del DER y según iteración e incremento</li> <li>Dividir el sistema en subsistemas (<i>si necesario</i>) <ul style="list-style-type: none"> <li>Agrupar funcionalidad según subsistemas <ul style="list-style-type: none"> <li>Refinar casos de uso</li> <li>Refinar diagrama preliminar de clases</li> <li>Elaborar diagramas de secuencia</li> </ul> </li> </ul> </li> <li>Representar subsistemas en arquitectura 3 capas (<i>si necesario</i>)</li> </ul>	<ul style="list-style-type: none"> <li>Estilos arquitectónicos [BCK98] <ul style="list-style-type: none"> <li>3 capas</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Listado descriptivo de las metas de diseño</li> <li>Estructura de la aplicación</li> <li>Arquitectura de la aplicación</li> </ul>
2. Diseño de la Interfaz Usuario/Sistema ( <i>si requerido</i> )	<ul style="list-style-type: none"> <li>Establecer el perfil de los usuarios</li> <li>Establecer perfil de tareas (<i>a partir de los casos de uso</i>)</li> <li>Establecer las características estéticas que deberá tener la interfaz gráfica de la aplicación <ul style="list-style-type: none"> <li>Establecer los fondos, colores, tipos de fuentes, etc.</li> </ul> </li> <li>Diseñar la estructura general de la interfaz U/S: <ul style="list-style-type: none"> <li>Elaborar el diagrama jerárquico de pantallas del sistema</li> <li>Definir las características que deben tener los ítems que componen las pantallas de la interfaz: menús, ventanas, íconos, enlaces, cuadros, cajas, etc.</li> </ul> </li> <li>Realizar las revisiones técnicas de la interfaz U/S según lo expresado en documentos DDR y DER</li> </ul>	<ul style="list-style-type: none"> <li>Técnicas de Utilidad (<i>usability</i>)</li> <li>Técnicas y estrategias de diseño de interfaces GUI</li> </ul>	<ul style="list-style-type: none"> <li>Diseño de pantallas</li> <li>Diagrama jerárquico de pantallas</li> </ul>
3. Diseño la BD ( <i>si requerido</i> )	<p>Realizar el diseño Conceptual</p> <ul style="list-style-type: none"> <li>Refinar el modelo preliminar de clases de objetos</li> <li>Definir o completar los atributos de cada clase de objetos de negocio</li> <li>Verificar el modelo con los requisitos</li> <li>Validar modelo con los usuarios</li> </ul> <p>Realizar el diseño implementable</p> <ul style="list-style-type: none"> <li>Convertir el modelo conceptual de la BD en un esquema relacional equivalente</li> <li>Verificar el esquema implementable con los requisitos relacionados</li> </ul> <p>Realizar el diseño Físico</p> <ul style="list-style-type: none"> <li>Establecer los índices de las tablas del diseño implementable</li> <li>Definir los derechos de acceso para cada tipo de usuario (<i>usuario final, programador, ABD</i>)</li> <li>Definir las reglas de integridad de la BD</li> </ul>	<ul style="list-style-type: none"> <li>Diagramas de clase en UML</li> <li>Modelado de Bases de Datos OO</li> <li>Procedimiento de conversión de diagramas de clase a esquemas de relación</li> <li>Modelado BD <i>Relacionales</i></li> <li>Revisión técnica (Inspección de Diseño)</li> <li>Procedimientos de diseño físico de BD <i>relacionales</i></li> </ul>	<ul style="list-style-type: none"> <li>Modelo conceptual integrado de la BD</li> <li>Esquema relacional o equivalente de la BD (<i>integrado y verificado</i>)</li> <li>Esquema físico de la BD</li> </ul>

4. Diseño de componentes o módulos de SW	<ul style="list-style-type: none"> <li>• Identificar Componentes o módulos</li> <li>• Identificar elementos funcionales – propios de la aplicación – implementación de casos de uso</li> <li>• Identificar elementos de interfaz U/S</li> <li>• Identificar elementos de acceso y manipulación de datos persistentes</li> <li>• Ubicar elementos en la arquitectura inicial predefinida –</li> <li>• Describir la interacción de componentes o módulos</li> <li>• Determinar las interfaces de cada componente</li> </ul>	<ul style="list-style-type: none"> <li>• UML <i>Components</i> [CHD01]</li> </ul>	<ul style="list-style-type: none"> <li>• Definición de componentes o módulos</li> <li>• Especificación de Interfaces</li> <li>• Arquitectura de Componentes o módulos del sistema</li> </ul>
5. Especificación del diseño	<ul style="list-style-type: none"> <li>• Integrar diagramas de subsistemas, interfaz, arquitectura y componentes o módulos y BD en Documento de Diseño</li> <li>• Definir los procedimientos de respaldo, recuperación y seguridad de la BD</li> </ul>	<ul style="list-style-type: none"> <li>• UML <i>Components</i> [CHD01]</li> </ul>	<ul style="list-style-type: none"> <li>• Documento de Diseño (DD) integrado y validado</li> <li>• Procedimientos de administración de la BD</li> </ul>



### 3.5. Proceso de Desarrollo - Proceso: Aprovisionamiento de componentes

Actividades	Tareas	Técnicas y Notaciones	Productos
1.Instalación de la plataforma de desarrollo	<ul style="list-style-type: none"> <li>• Seleccionar, adquirir y/o preparar la plataforma o infraestructura de software requerida para desarrollar el sistema</li> <li>• Instalar la plataforma de desarrollo (servidores web, de aplicaciones, SMBD)</li> </ul>	<ul style="list-style-type: none"> <li>• Manuales y procedimientos de instalación de software</li> </ul>	<ul style="list-style-type: none"> <li>• Plataforma de desarrollo instalada</li> </ul>
2.Adquisición de Componentes – <i>si requeridos</i>	<ul style="list-style-type: none"> <li>• Buscar componentes que puedan ser adquiridos de terceros (abiertos o propietarios) o en librerías propias de la organización</li> <li>• Adquirir componentes</li> </ul>	<ul style="list-style-type: none"> <li>• Búsqueda de componentes abiertos o comerciales (P. ej., COTS)</li> </ul>	<ul style="list-style-type: none"> <li>• Componentes adquiridos</li> </ul>
3.Adaptación de Componentes <i>-si requerido</i>	<ul style="list-style-type: none"> <li>• Buscar componentes en repositorios locales (internos) o de terceros (externos)</li> <li>• Adaptar los componentes mediante su modificación interna o el uso de envoltorios (<i>wrappers</i>)</li> </ul>	<ul style="list-style-type: none"> <li>• Reutilización caja blanca</li> <li>• Envoltorios (<i>Wrapping</i>)</li> </ul>	<ul style="list-style-type: none"> <li>• Componentes adaptados</li> </ul>
4.Desarrollo de Componentes o módulos	<ul style="list-style-type: none"> <li>• Desarrollar aquellos componentes o módulos que no pudieron ser localizados o adquiridos. Partiendo de la especificación de cada componente: <ul style="list-style-type: none"> <li>○ Elaborar el diseño detallado de cada operación de cada interfaz del componente</li> <li>○ Codificar las operaciones del componente</li> <li>○ Elaborar interface del componente</li> <li>○ Desplegar el componente en la plataforma seleccionada</li> <li>○ Diseñar y ejecutar las pruebas de unidad del componente</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Diseño de algoritmos</li> <li>• Refinamiento paso-a-paso</li> <li>• Pseudo-código</li> <li>• Estándares de codificación</li> <li>• Estrategias de pruebas de unidad</li> </ul>	<ul style="list-style-type: none"> <li>• Componentes desarrollados</li> </ul>
5. Diseño y ejecución de pruebas de componentes o módulos	<ul style="list-style-type: none"> <li>• Realizar pruebas funcionales para cada uno de los componentes o módulos adquiridos, adaptados y/o desarrollados <ul style="list-style-type: none"> <li>○ Preparar los datos y mecanismos de prueba</li> <li>○ Preparar el ambiente de pruebas</li> <li>○ Ejecutar las pruebas funcionales de cada componente</li> </ul> </li> <li>• Depurar los errores encontrados durante las pruebas funcionales de cada componente o módulo</li> </ul>	<ul style="list-style-type: none"> <li>• Estrategias de pruebas caja negra: <ul style="list-style-type: none"> <li>○ Clases de equivalencia</li> <li>○ Análisis de valores límites</li> </ul> </li> <li>• Técnicas de pruebas de hilos (<i>thread testing</i>)</li> <li>• Depuración de errores</li> </ul>	<ul style="list-style-type: none"> <li>• Especificaciones de casos de prueba</li> <li>• Componentes y/o módulos probados y depurados</li> </ul>

### 3.6. Procesos de Desarrollo - Proceso: Ensamblaje del Sistema de software

Actividades	Tareas	Técnicas y Notaciones	Productos
1.Construcción de la Interfaz U/S	<ul style="list-style-type: none"> <li>Ensamblar la capa de presentación con los componentes o elementos de SW de la interfaz U/S               <ul style="list-style-type: none"> <li>Codificar e integrar los componentes o elementos del lado del cliente</li> </ul> </li> <li>Diseñar y Ejecutar las Pruebas de la Interfaz U/S               <ul style="list-style-type: none"> <li>Determinar los aspectos de la interfaz U/S que deben probarse</li> <li>Realizar prueba de la interfaz U/S                   <ul style="list-style-type: none"> <li>Preparar los datos y mecanismos de prueba</li> <li>Preparar el ambiente de pruebas</li> <li>Ejecutar las pruebas de la interfaz U/S</li> </ul> </li> </ul> </li> <li>Depurar los errores encontrados</li> </ul>	<ul style="list-style-type: none"> <li>Técnicas de construcción de software</li> <li>Técnicas y estrategias de pruebas de interfaces gráficas</li> <li>Depuración de errores</li> </ul>	<ul style="list-style-type: none"> <li>Especificaciones de casos de prueba</li> <li>Interfaz U/S probada</li> </ul>
2.Ensamblaje de Componentes o módulos de la aplicación – capa de negocios	<ul style="list-style-type: none"> <li>Ensamblar la capa de lógica de negocios – subsistemas - componentes o módulos de la aplicación que la integran</li> <li>Ejecutar las Pruebas de Integración               <ul style="list-style-type: none"> <li>Definir los criterios y técnicas de pruebas de integración de componentes o módulos</li> <li>Realizar casos de prueba de integración de componentes o módulos                   <ul style="list-style-type: none"> <li>Preparar los datos y mecanismos de prueba</li> <li>Preparar el ambiente de pruebas</li> <li>Ejecutar las pruebas de integración de componentes o módulos</li> </ul> </li> </ul> </li> <li>Depurar los errores encontrados</li> </ul>	<ul style="list-style-type: none"> <li>Despliegue de componentes o módulos en servidores de aplicaciones</li> <li>Técnicas y estrategias de pruebas de integración de componentes o módulos</li> <li>Depuración de errores</li> </ul>	<ul style="list-style-type: none"> <li>Especificaciones de casos de prueba</li> <li>Capa de Lógica de negocios de la aplicación probada</li> </ul>
3.Construcción de la BD ( <i>si requerido</i> )	<ul style="list-style-type: none"> <li>Crear la base de datos usando los esquemas implementables diseñados en el proceso anterior y según el SMBD seleccionado</li> <li>Diseñar y Ejecutar las Pruebas de la BD               <ul style="list-style-type: none"> <li>Realizar casos de prueba de la BD                   <ul style="list-style-type: none"> <li>Definir los aspectos de la BD que deben probarse</li> <li>Preparar los datos y mecanismos de prueba</li> <li>Preparar el ambiente de pruebas</li> <li>Ejecutar las pruebas de la BD</li> </ul> </li> </ul> </li> <li>Depurar los errores encontrados</li> </ul>	<ul style="list-style-type: none"> <li>Creación de BD relacional</li> <li>Técnicas y estrategias de pruebas de bases de datos</li> <li>Depuración de errores</li> </ul>	<ul style="list-style-type: none"> <li>Especificaciones de casos de prueba</li> <li>Base de datos probada</li> </ul>
4.Pruebas de la Integración de las Capas de la arquitectura	<ul style="list-style-type: none"> <li>Realizar casos de prueba de integración de capas               <ul style="list-style-type: none"> <li>Definir los criterios y técnicas de pruebas de integración de las tres capas de la aplicación</li> <li>Preparar los datos y mecanismos de prueba</li> <li>Preparar el ambiente de pruebas</li> <li>Ejecutar las pruebas de integración de capas</li> </ul> </li> <li>Depurar los errores encontrados</li> </ul>	<ul style="list-style-type: none"> <li>Técnicas y estrategias de pruebas de aplicaciones distribuidas</li> <li>Depuración de errores</li> </ul>	<ul style="list-style-type: none"> <li>Especificaciones de casos de prueba</li> <li>Aplicación integrada y probada (desplegada en la plataforma de desarrollo)</li> </ul>

### 3.7. Procesos de Desarrollo - Proceso: Pruebas del Sistema de Software

Actividades	Tareas	Técnicas y Notaciones	Productos
1. Realización de las Pruebas del Sistema	<ul style="list-style-type: none"> <li>Definir mecanismos de pruebas</li> <li>Preparar casos de prueba</li> <li>Realizar las pruebas funcionales del sistema (aplicación integrada) <ul style="list-style-type: none"> <li>Ejecutar las pruebas funcionales</li> </ul> </li> <li>Realizar las pruebas no funcionales del sistema <ul style="list-style-type: none"> <li>Ejecutar las pruebas no-funcionales</li> </ul> </li> <li>Realizar las pruebas de aceptación <ul style="list-style-type: none"> <li>Ejecutar las pruebas de aceptación</li> </ul> </li> </ul> <p>Reportar los errores encontrados en las pruebas</p>	<ul style="list-style-type: none"> <li>Estrategias de pruebas funcionales y no funcionales</li> </ul>	<ul style="list-style-type: none"> <li>Mecanismos de pruebas</li> <li>Casos de pruebas</li> <li>Especificaciones de casos de prueba</li> <li>Informe de incidentes de prueba</li> </ul>
2. Corrección de errores	<ul style="list-style-type: none"> <li>Corregir los errores detectados en las pruebas funcionales y no-funcionales</li> <li>Realizar pruebas de regresión para asegurar que las correcciones no introducen nuevos errores</li> </ul>	<ul style="list-style-type: none"> <li>Depuración (<i>debugging</i>)</li> </ul>	<ul style="list-style-type: none"> <li>Aplicación probada y depurada</li> </ul>

### 3.8. Procesos de Desarrollo - Proceso: Entrega del Sistema de Software

Actividades	Tareas	Técnicas y Notaciones	Productos
1. Instalación de la Aplicación	<p>Planificar la instalación</p> <ul style="list-style-type: none"> <li>Definir las estrategias de migración a la nueva aplicación</li> <li>Determinar actividades de la instalación de la aplicación</li> <li>Instalar la plataforma de Hardware/Software requerida para operar el sistema (<i>si no está instalada</i>)</li> <li>Desplegar la aplicación en los diferentes servidores de la plataforma de operación; carga inicial de datos (<i>si se requiere</i>) <ul style="list-style-type: none"> <li>Preparar los datos de carga inicial de la BD</li> <li>Actualizar la BD (<i>si requerido</i>)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Técnicas y herramientas de planificación</li> <li>Instructivos de despliegue de aplicaciones proporcionados por el fabricante</li> <li>Técnicas de migración de datos</li> </ul>	<ul style="list-style-type: none"> <li>Plan de Instalación</li> <li>Plataforma de Operación (H/S) instalada</li> <li>Aplicación instalada</li> <li>BD actualizada</li> </ul>
2. Diseño y Ejecución de Pruebas de Instalación	<ul style="list-style-type: none"> <li>Definir los aspectos de la instalación que deben probarse</li> <li>Diseñar los procedimientos y casos de prueba de instalación</li> <li>Preparar los datos y mecanismos de prueba</li> <li>Ejecutar las pruebas de instalación</li> <li>Corregir los errores encontrados</li> <li>Identificar cambios y ajustes finales</li> <li>Entregar el sistema al cliente</li> </ul>	<ul style="list-style-type: none"> <li>Técnicas y estrategias de pruebas de instalación</li> <li>Depuración de errores</li> <li>Control de cambios en las especificaciones del software</li> </ul>	<ul style="list-style-type: none"> <li>Especificaciones de casos de prueba</li> <li>Informe de incidentes de prueba</li> <li>Aplicación instalada probada</li> </ul>
4. Ensamblaje de Documentación	<ul style="list-style-type: none"> <li>Ensamblar los documentos parciales del producto de SW – elaborados durante todo el desarrollo del proyecto</li> </ul>	<ul style="list-style-type: none"> <li>Técnicas de elaboración de documentos técnicos</li> </ul>	<ul style="list-style-type: none"> <li>Documentos o manuales de la aplicación</li> </ul>
5. Capacitación de Usuarios	<ul style="list-style-type: none"> <li>Preparar ambiente y material de capacitación</li> <li>Conducir capacitación de usuarios</li> </ul>	<ul style="list-style-type: none"> <li>Técnicas de capacitación</li> </ul>	<ul style="list-style-type: none"> <li>Material de capacitación</li> </ul>

**Conclusiones:**

- El Método *White-Watch* (W\_Watch)
  - propone un modelo de procesos que balancea la producción de especificaciones de producto que se transforman en la medida que se avanza en el proceso, con la disponibilidad en corto tiempo de versiones parciales y operativas del producto.
  - logra dedicar más tiempo a la producción y puesta a punto del software – versiones del producto gracias al apoyo brindado por las herramientas automatizadas de desarrollo y documentación.
- Un Método de Desarrollo de software es más que una Metodología de desarrollo, pues describe, además del proceso que debería seguirse para la solución del problema, el producto final (el software y sus componentes parciales) y el equipo de trabajo que ejecutará el proceso de desarrollo.
- Adaptar una propuesta metodológica requiere de experiencia en el área – desarrollo de software, y de conocimiento del dominio – tipo de aplicaciones y tecnología disponible, del contexto organizacional y de gestión de proyectos de software; todo esto para lograr la instanciación y/o configuración del método W\_Watch, más adecuada a las necesidades particulares de cada proyecto.