

Capítulo 6

Conclusiones

6.1. Sobre el Modelo

- Utilización de la ingeniería de software como mecanismo de aplicación y evaluación de la eficiencia y calidad operacional de un sistema de función crítica, visto como la definición de criterios de operación bajo condiciones y límites establecidos por el sistema y por las características externas del medio externo.
- En el desarrollo de productos de software las etapas de análisis de requerimientos y diseño toma gran parte del tiempo del proyecto. El modelo planteado en este proyecto pretende establecer unos parámetros de diseño generales que permitan agilizar la implementación de proyectos tipo sistemas de control por software, cuya base común es el procesamiento de señales digitales en busca de comportamientos de interés (*caracterización de señales*).
- La utilización de un ciclo de vida específico para el desarrollo de software,

basado en las condiciones del tipo de problemas a tratar, constituye uno de los alcances notables del modelo ofrecido. El ciclo de vida contempla la noción de fases generales que constituyen un marco de situación, estableciendo fases de solución para un subproblema concreto.

- Con el constante desarrollo e innovación de las tecnologías utilizadas en las implementaciones de software, es deseable tener un modelo no dependiente de mecanismos, métodos y plataformas específicas, adecuándolo a necesidades y ambientes particulares. Si bien se han utilizado conceptos de paradigmas como el de desarrollo orientado a objetos o sistemas en tiempo real, el modelo ha buscado generalizarse para que su interpretación pueda hacerse según condiciones singulares de los problemas a tratar.
- La consideración de un mecanismo para realizar la gestión del riesgo hace parte de los principios técnicos para el desarrollo de proyectos de ingeniería. A nivel de la Ingeniería de software y del modelo planteado, la gestión actúa como instrumento para el control de calidad y como guía para conocer las limitaciones y características del ciclo de vida.

6.2. Sobre la herramienta

La implementación del software, constituye una aplicación y comprobación simple del modelo descrito. Desde el punto de vista técnico, se han utilizado conceptos básicos como :

- Modularización que favorece el depuramiento y codificación de la aplicación durante el proceso de implementación. Durante el mantenimiento del

software, favorece en el entendimiento y adición de características.

- Uso de protocolos y estándares abiertos para comunicaciones y programación, facilitando el paso de mensajes y datos entre módulos y métodos internos y externos.
- Diversos lenguajes de programación, realizando implementación de algunos módulos en lenguajes que ofrezcan mejores condiciones que otros en ambientes particulares de uso.
- Siguiendo las actividades sugeridas en el modelo, se presentan beneficios al tener una estructura homogénea y clara reflejada en el diseño y la codificación del producto de software.
- La aplicación de las pruebas durante las etapas, permiten agilizar el proceso de depuramiento, ya que no es necesario tener toda la aplicación funcionando, sino que se diagnostica el funcionamiento de componentes más simples y recientemente codificados.
- Utilización de estructuras de datos de varios ordenes: la estructura DSRC como estructura nativa del lenguaje de programación es rápida en uso. la estructura XMLDSRC como estructura de alto nivel, basada en etiquetas facilita el transporte entre aplicaciones de diversos lenguajes y arquitecturas.

6.3. Conclusiones generales

- Se realizó la construcción de un modelo funcional para la caracterización de señales.

- Establecimiento de mecanismos para el procesamiento adaptativo, basado en el comportamiento dinámico de las señales y las características generales del procesamiento de señales digitales.
- Implementación de un sistema de control por software, dando la categorización de sistema de control por sus condiciones de operación, construcción de alto nivel y con la flexibilidad para personalizar, propio de los sistemas de software.
- Elaboración de una herramienta para la detección de eventos, independiente de un sistema de adquisición específico.
- Utilización de la plataforma Linux y el modelo de desarrollo de código abierto (OpenSource), para la implementación de software robusto siguiendo principios de diseño de un sistema crítico y de bajo costo de construcción en software y hardware.
- Factibilidad de aplicación del modelo y desarrollo técnico a otros tipos de problema que requieran procesamiento de señales, específicamente detección de cambios o eventualidades sobre señales digitales.
- Con la propuesta de un modelo 'portable' en cuanto a análisis y diseño se observa la factibilidad de enfocarlo, bajo cierto nivel de abstracción a problemas similares en estructura, puntualizando en los detalles propios del sistema.
- El modelo considera aspectos de gestión de proyectos, análisis, diseño del software, gestión de riesgos, control de calidad, entre otros.

- Bajo la idea de cómo se concibe este modelo, se puede evaluar si es posible aplicar el concepto general a nuevos campos o ambientes, definiendo nuevos modelos generales que puedan ser correspondientes a tipos de problemas específicos. Parte de la utilidad que tendría, es el poder agilizar la producción de software, sin comprometer la no realización de actividades como el análisis y diseño, poniendo en riesgo el desarrollo del proyecto.