**Análisis y Diseño de Aplicaciones**

*Integrantes:*

Lucas Rivera Matias Rodriguez

Alex Serpa Timothy de Souza

Ingeniería de software:

Es el área de la Ingeniería que ofrece metodos y tecnicas para desarrollar y mantener software. Un ingeniero de software es un programador que está especializado en un determinado lenguaje de programación. Una de sus funciones es hacer un análisis de la futura arquitectura y de sus distintas piezas de software.

La creación de software es un proceso creativo que la I.S. trata de sistematizar con el fin de acotar el riesgo del fracaso en la obtención del objetivo creativo.

La I.S. se puede considerar como la ingeniería aplicada al software, la aplicación de la forma más eficiente para la obtención de resultados óptimos.

Implicaciones: La I.S. afecta a la economía y las sociedades.

Económicamente: En los años 90, La I.S afectó mucho la economía, contribuyó con mas de 1 Billón de U$S en el crecimiento económico y en la productividad. Hoy en día, el software contribuye cada vez más en el crecimiento económico y productivo en el mundo.

Socialmente: La I.S. cambia la cultura del mundo y mejora la calidad de los servicios como salud, educación, departamento de bomberos, dependencias gubernamentales y otros servicios sociales. También juega un papel muy importante en la globalización, gracias a que es la encargada de desarrollar todos los medios de comunicaciones y redes sociales que usamos día a día.

Metodología: Hace décadas ha sido un objetivo encontrar procesos y metodologías. Para simplificar y mejorar la productividad en el desarrollo y la calidad del software.

Etapas del Proceso:

La I.S. utiliza una metodología específica para realizar numerosas tareas, dentro de esta metodología se encuentran etapas como las siguientes:

Análisis de requerimientos: Es la primera etapa, consiste en extraer los requisitos y requerimientos que serán necesarios para el software, para hacer esto se necesita de habilidad y experiencia en la I.S ya que es necesario reconocer requerimientos incompletos, ambiguos o contradictorios.

La captura, análisis y especificación de requerimientos es una parte crucial, afectará en gran medida el logro de los objetivos finales.

Especificación: Describe el comportamiento que deberá tener el software una vez desarrollado, para hacer estas es necesario hacer una entrevista con el cliente, analizar los requisitos que el quiere que el programa cumpla, y expresarlos en un documento.

Entre las técnicas utilizadas para la especificación de requisitos se encuentran:

• Casos de Uso (Riguroso y formal).

• Historias de usuario (Ágil e informal).

Arquitectura: El Arquitecto de Software es la persona que se encarga del diseño de componentes de una aplicación, aportar principalmente soluciones tecnológicas, entre otras. Es una tarea que requiere de planeación, ya sea a nivel de infraestructura de red y hardware, o de Software.

Programación: Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no necesariamente es la que demanda mayor trabajo y ni la más complicada. La complejidad y la duración de esta etapa está íntimamente relacionada al o a los lenguajes de programación utilizados, así como al diseño previamente realizado.

Prueba: Consiste en comprobar que el software realice correctamente las tareas indicadas. Una técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo.

Documentación: Todo lo concerniente a la documentación del propio desarrollo del software y de la gestión del proyecto, pasando por

modelaciones (UML), diagramas, pruebas, manuales de usuario, manuales técnicos, etc; todo con el propósito de

eventuales correcciones, usabilidad, mantenimiento futuro y ampliaciones al sistema.

Mantenimiento: Mantener y mejorar el software es necesario para arreglar errores descubiertos y cumplir nuevos requisitos. Esto puede llevar más

tiempo incluso que el desarrollo inicial del software. Alrededor de ⅔.

La parte más duradera es la de actualizar el programa conforme vaya necesitando funciones nuevas.

Naturaleza de la I.S:

La Ingeniería de Software se relaciona con varios campos en diferentes formas:

Matemáticas: Los programas tienen muchas propiedades matemáticas. Por ejemplo la corrección y la complejidad de muchos algoritmos. El uso de matemáticas en la I.S se le llama métodos formales.

Creación: Los programas son construidos en una secuencia de pasos (Metodología). El hecho de definir propiamente y llevar a cabo estos pasos, es necesario para mejorar la productividad de los desarrolladores y la calidad final de los programas.

Gestión de Proyectos: Se requiere gestión de proyectos. Hay presupuestos y establecimiento de tiempos. Gente para liderar. Recursos (espacio de oficina, computadoras) por adquirir. Todo esto encaja apropiadamente con la visión de la Gestión de Proyectos.

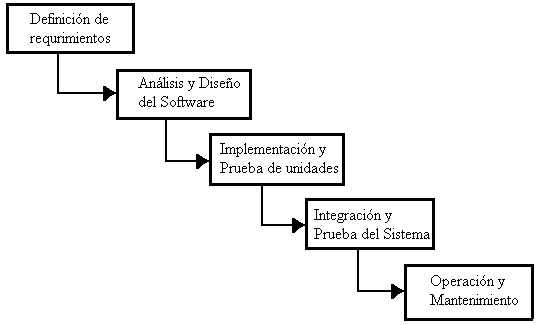
Arte: Los programas contienen muchos elementos artísticos. Las interfaces de usuario, la codificación, etc. Incluso la decisión para un nombre de una variable o una clase.

Responsabilidad: La responsabilidad en la Ingeniería del Software es muy importante, sobre todo porque al estar los sistemas informáticos fuertemente caracterizados por su complejidad, es difícil apreciar sus consecuencias.

En la Ingeniería del Software la responsabilidad será compartida por un grupo grande de personas, desde el ingeniero de requisitos, hasta el arquitecto software, y contando con el diseñador, o el encargado de realizar las pruebas. Por encima de todos ellos destaca el director del proyecto. El software demanda una clara distribución de la responsabilidad entre los diferentes roles.

Modelos de proceso o ciclo de vida:

Modelo en cascada: El modelo en cascada puro difícilmente se utilice tal cual, pues esto implicaría un previo y absoluto conocimiento de los requisitos, la no volatilidad de ellos y etapas subsiguientes libres de errores, sólo podría ser aplicable a escasos y pequeños desarrollos de sistemas, aun así no recomendable. En este proceso de vida del software, si pasas a la etapa siguiente, no hay retorno para arreglar fallos o modificar el software. El mayor problema es que el software es de carácter evolutivo, cambiante y difícilmente libre de errores, tanto durante su desarrollo como durante su vida operativa, por eso este modelo es muy difícil de ver.



Sin embargo, el modelo cascada en algunas de sus variantes es uno de los actualmente más utilizados, por su eficacia y simplicidad, más que nada en software de pequeño y algunos de mediano porte.

Modelo en cascada realimentado: Es una de las variantes del Modelo en cascada, esta permite que haya cambios (realimentaciones) entre etapas, permitiendo corregir errores de previas etapas, hacer cambios o evoluciones durante el ciclo de vida.

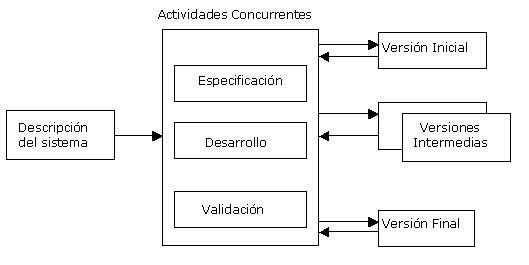
Modelos Evolutivos: Los requisitos del usuario y del producto suelen cambiar conforme se desarrolla el mismo. Las fechas de mercado y la competencia hacen que no sea posible esperar a poner en el mercado un producto absolutamente completo, por lo que se debe introducir una versión funcional limitada de alguna forma para aliviar las presiones competitivas.

En esas u otras situaciones similares los desarrolladores necesitan modelos de progreso que estén diseñados para acomodarse a una evolución temporal o progresiva. Los evolutivos son modelos iterativos, permiten desarrollar versiones cada vez más completas y complejas.

En los modelos evolutivos, el iterativo incremental y el espiral son dos de los más conocidos y usados.

Modelo Iterativo Incremental: Permite la entrega de versiones parciales a medida que se va construyendo el producto final.

La Descripción del Sistema es esencial para especificar y confeccionar los distintos incrementos. Las actividades concurrentes (Especificación, Desarrollo y Validación) sintetizan el desarrollo pormenorizado de los incrementos, que se hará posteriormente.



El incremental es un modelo de tipo evolutivo que utiliza varios ciclos Cascada realimentados aplicados repetidamente. Cada ciclo cascada es aplicado para la obtención de un incremento, estos últimos se van integrando para obtener el producto final completo. Cada incremento es un ciclo Cascada Realimentado.

Existen actividades de desarrollo (para cada incremento) que son realizadas en paralelo, por ejemplo, en la figura, mientras se realiza el diseño detalle del primer incremento ya se está realizando en análisis del segundo.

