

ESTIMACIÓN DE ESFUERZO, COSTO Y TIEMPO

Utilizar los Casos de Uso que permite documentar a los requisitos de una aplicación en términos de Actores y Casos de Uso.

Como se ha explicado en anteriores documentos, un Actor típicamente representa a un usuario humano o a otra aplicación que interactúa con la aplicación bajo Desarrollo.

Un Caso de Uso representa un requisito de la Aplicación bajo análisis, relata de forma secuencial las acciones que uno o más actores llevan a cabo en el sistema para obtener un resultado de valor significativo.

Si bien los Casos de Uso permiten especificar la funcionalidad de una aplicación bajo análisis, no permiten, por sí mismos, efectuar una estimación del tamaño que tendrá la aplicación o del esfuerzo que tomaría implementar.

Para la estimación del tamaño de una aplicación a partir de sus requisitos, una de las técnicas más difundidas es el Análisis de Puntos de Función.

Esta técnica permite cuantificar el tamaño de un sistema en unidades independientes del lenguaje de programación, las metodologías, plataformas y/o tecnologías utilizadas.

Por otro lado, el SEI (del inglés, Software Engineering Institute) propone desde hace algunos años un método para la estimación del esfuerzo llamado COCOMO II.

Este método está basado en ecuaciones matemáticas que permiten calcular el esfuerzo a partir de ciertas métricas de tamaño estimado, como el Análisis de Puntos de Función y las líneas de código fuente (en inglés SLOC, Source Line Of Code).

Existe una relación natural entre los Puntos de Función y los Casos de Uso. Los Puntos de Función permiten estimar el tamaño del software a partir de sus requisitos, mientras que los Casos de Uso permiten documentar los requisitos.

Ambos tratan de ser independientes de las tecnologías utilizadas para la implementación. En etapas tempranas del ciclo de vida, se identifican los Actores y los Casos de Uso de la aplicación y se documenta cada uno de estos mediante una breve descripción.

Utilizando el Análisis de Puntos de Función a estos Casos de Uso, se podrá obtener una estimación grosera del tamaño y a partir de esta del esfuerzo.

Esta estimación es bastante imprecisa debido, principalmente, a la escasa información que se tiene sobre el software al principio de un proyecto, pero permitirá obtener una idea del esfuerzo necesario para llevar adelante el mismo, y podrá ser refinada a medida que se obtenga más información.

Se recalca que es una estimación, lo más importante de esta actividad es que se podrá realizar un plan para el desarrollo de software.

La Planificación está muy relacionada con las actividades de la estimación, para determinar las actividades dentro de la planificación es muy necesario conocer el tiempo, costo y el esfuerzo que se necesitarán para llevar a cabo el desarrollo.

Pressman señala lo siguiente: Para llevar a cabo un buen proyecto de desarrollo de software, debemos comprender el ámbito del trabajo a realizar, las tareas a ejecutar, las referencias a tener en cuenta, la agenda a seguir, el esfuerzo a emplear y los recursos requeridos.

Con la descripción y estudio de los casos de uso se determina el ámbito del trabajo, las tareas a ejecutar y las referencias a tener en cuenta.

Con la estimación con ayuda de los Puntos Función y los Casos de Uso, se llega a determinar los recursos y el esfuerzo que se van a necesitar.

Dentro de estos documentos que describen el desarrollo de una aplicación no se llega a tocar el punto de planificación del proyecto, la agenda de las actividades que se deben realizar para llevar a cabo el desarrollo.

Pero, todo en la vida se debe planificar, si no se llega a obtener una planificación no se podrá hacer, no se tendrá una estrategia y no se tendrá un proyecto.

El plan de proyecto debe tener en cuenta el alcance, tareas, calidad, métricas, cronogramas y disponibilidad de recursos.

El desarrollo de software cumple un ciclo de vida como todo producto tangible, el ciclo de vida del proyecto de software se puede determinar en

cinco fases de proceso que determina la Guía del PMBOK y son: Proceso de Inicialización, Planificación, Ejecución, Seguimiento y Control y el Proceso de Cierre.

En un próximo conjunto de documentos se hablará de la Gestión de Proyectos de Software.

Para entender con mayor facilidad se define los siguientes conceptos:

- **Esfuerzo:** Tiempo que necesita una persona para trabajar en el desarrollo del proyecto (hombres/mes, hombres/días, hombres/horas).
- **Tiempo:** duración total del proyecto.
- **Cantidad de Personas:** Recursos necesarios para desarrollar el software.
- **Costo:** cantidad de dinero que se necesita para llevar a cabo el desarrollo de software.
- **Transacciones:** Está representada por uno o más pasos del flujo de eventos principal del Caso de Uso, pudiendo existir más de una transacción dentro del mismo Caso de Uso. Los flujos de eventos alternativos dentro del Caso de Uso ayudan a clarificar las transacciones.

ESTIMACIÓN POR PUNTOS DE CASOS DE USO.

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado por muchos otros autores.

Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de “pesos” a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

El primer paso es calcular los Puntos de Casos de Uso sin Ajustar. Este valor se calcula con la siguiente ecuación:

1.- Puntos de Casos de Uso sin Ajustar:

$$UUCP = UAW + UUCW$$

donde:

- **UUCP:** Puntos de Casos de Uso sin ajustar
- **UAW:** Factor de Peso de los Actores sin ajustar
- **UCW:** Factor de Peso de los Casos de Uso sin ajustar

El Factor de Peso de los Actores (UAW) sin ajustar se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece teniendo en cuenta:

- Si se trata de una persona o de otro sistema
- La forma en la que el actor interactúa con el sistema.

Los criterios se muestran en la siguiente tabla:

Tipo de Actor	Descripción	Factor de Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface)	1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica	3

Tabla 2. Posibles Factores de Peso de los Actores

El Factor de Peso de los Casos de Uso sin Ajustar (UUCW) se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos.

La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia.

Los criterios se muestran en la siguiente tabla:

Tipo de Caso de Uso	Descripción	Factor de Peso
Simple	El caso de uso tiene de 1 a 3 transacciones	5
Medio	El caso de uso tiene de 4 a 7 transacciones	10
Complejo	El caso de uso contiene más de 8 transacciones	15

Tabla 3. Posibles Factores de Peso para los Casos de Uso

Aplicando el análisis al ejemplo de desarrollo que llevamos en estos documentos, se realiza el cálculo de Los Puntos de Casos de Uso sin Ajustar.

Análisis de los Actores para encontrar el Factor de Peso de Actores sin Ajustar:

Actor	Factor Peso
Cliente	3
Recepcionista	3
Jefe de Cocina	3
Administrador	3

$UAW = 3+3+3+3 = 12$ (Peso de los Actores sin Ajustar)

Análisis de los Casos de Uso para encontrar el Factor de Peso de Casos de Uso sin Ajustar:

El análisis se debe hacer a cada caso de uso y pensar cuantas transacciones tiene cada uno de ellos.

Casos de Uso	Factor de Peso
Reservar Habitación	(Una transacción) 5
Confirmar Reserva	(Dos transacciones) 5
Salir del Hotel	(Cuatro Transacciones) 10
Cambiar Contraseña	(Una Transacción) 5
Autenticar Empleado	(Una Transacción) 5
Registrar Solicitud de Servicio a la Habitación	(Tres Transacciones) 5
Registrar Solicitud de Servicio Básico	(Tres Transacciones) 5
Gestionar Empleados	(Siete Transacciones) 10
Gestionar Bebidas	(Siete Transacciones) 10
Gestionar Comidas	(Siete Transacciones) 10
Gestionar Habitación	(Siete Transacciones) 10

UUCW= 5+5+10+5+5+5+5+10+10+10+10 = 80 (Peso de Casos de Uso sin Ajustar)

Finalmente, los Puntos de Casos de Uso sin ajustar sería:

$$UUCP = 12 + 80 = 92$$

Nos falta calcular varias cosas antes de obtener el esfuerzo.

2.- Puntos de Casos de Uso ajustados:

Una vez que se tiene los Puntos de Casos de Uso sin Ajustar se debe ajustar este valor mediante la siguiente ecuación:

UCP = UUCP x TCF x EF donde:

- **UCP:** Puntos de Casos de Uso ajustados
- **UUCP:** Puntos de Casos de Uso sin ajustar
- **TCF:** Factor de complejidad técnica
- **EF:** Factor de ambiente

TCF: Factor de Complejidad Técnica:

Para calcular el Factor de Complejidad Técnica se debe definir varios Factores que influyen en la complejidad técnica, son los siguientes:

Factor	Descripción	Peso
T1	Sistema Distribuido	2
T2	Objetivos de Comportamiento o tiempo de respuesta	1
T3	Eficacia del Usuario Final	1
T4	Procedimiento Interno Complejo	1
T5	El código debe ser reutilizable	1
T6	Facilidad de instalación	0.5
T7	Facilidad de Uso	0.5
T8	Portabilidad	2
T9	Facilidad de Cambio	1
T10	Concurrencia	1
T11	Incluye objetos especiales de seguridad	1
T12	Provee Acceso directo a terceras partes	1
T13	Se requiere facilidades especiales de entrenamiento a usuarios	1

Tabla 3. Factores que influyen en la Complejidad Técnica

- Sistema Distribuido. La aplicación a desarrollar será distribuida si varios módulos estarán en varios lugares.
- Objetivos de Comportamiento o tiempo de respuesta. Si es necesario que el sistema de una respuesta en un espacio de tiempo mínimo. Si es en un entorno Web, el cargado entre paginas no sea muy lento.
- Eficacia del Usuario Final. El usuario debe tener varios caminos para realizar su trabajo incrementando su eficacia.
- Procedimiento Interno Complejo. Si la codificación será compleja y requerirá de investigación para realizarla o para optimizarla.
- El código debe ser reutilizable. Si varios módulos o componentes deben poder ser utilizados en otras aplicaciones.
- Facilidad de Instalación. Si se debe crear instaladores para la cómoda configuración de la aplicación.
- Facilidad de uso. La aplicación debe ser fácil de aprender, recordar, visible, entendible, etc
- Portabilidad. La aplicación está desarrollada para facilitar el traslado de la tecnología a otra.
- Facilidad de Cambio. La aplicación debe estar implementada de manera que sea fácil detectar defectos y realizar los cambios para eliminarlos.
- Concurrencia. Si la aplicación será utilizada por un conjunto de personas grande debe comportarse de manera óptima
- Incluye objetivos especiales de seguridad. Si va a ser necesario implementar parte de la seguridad para los datos o el acceso a la aplicación.
- Provee acceso a terceras partes. Si la aplicación será utilizada por otras aplicaciones.
- Se requiere facilidades especiales de entrenamiento a usuarios. Si se debe planificar un entrenamiento para que la aplicación sea utilizada.

Valores asignados:

Este coeficiente se calcula mediante la cuantificación de los factores explicados anteriormente, que determinan la complejidad técnica del sistema.

Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

En la siguiente tabla se muestra el significado y el peso de cada uno de estos factores:

El Factor de Complejidad Técnico se calcula mediante la ecuación:

$$TCF = 0.6 + 0.01 * S(\text{Peso}_i * \text{ValorAsignado}_i)$$

Para nuestro ejemplo el Factor de Complejidad Técnica tendrá el valor:

Factor	Descripción	Valor Asignado	Comentario
T1	Sistema Distribuido	2	La aplicación utilizará varios Servicios Web
T2	Objetivos de Comportamiento o tiempo de respuesta	4	La aplicación debe proporcionar una alta rapidez de respuesta a los Cliente
T3	Eficacia del Usuario Final	4	Debe asegurarse que la aplicación proporcione una alta eficacia para los usuarios
T4	Procedimiento Interno Complejo	2	No muchos cálculos complejos
T5	El código debe ser reutilizable	5	El código debe ser reutilizable, los cual permita ser utilizado en otras aplicaciones
T6	Facilidad de instalación	2	Escasos requisitos de facilidad de instalación.
T7	Facilidad de Uso	5	Es muy necesario crear interfaz que sean fáciles de aprender, memorizar, que sean agradables y comprensibles.
T8	Portabilidad	2	Es necesario que sea compatible con los distintos navegadores Web existentes
T9	Facilidad de Cambio	1	Se debe garantizar la utilización de un estándar para la codificación
T10	Concurrencia	3	La concurrencia será relativamente fluida

T11	Incluye objetivos especiales de seguridad	2	Seguridad Baja
T12	Provee Acceso directo a terceras partes	1	Los Usuarios Web tienen acceso
T13	Se requiere facilidades especiales de entrenamiento a usuarios	1	

$$TCF = 0.6 + 0.01 * (2*2 + 1*4 + 1*4 + 1*2 + 1*5 + 0.5*2 + 0.5*5 + 2*2 + 1*1 + 1*3 + 1*2 + 1*1 + 1*1)$$

$$TCF = 0.945$$

Factor Ambiente (EF):

Otro factor importante que debe calcularse es el Factor Ambiente (EF). Este factor determinar las habilidades.

Y entrenamiento del grupo involucrado en el desarrollo tiene un gran impacto en la estimación de tiempo. El cálculo de este factor es similar al de Factor de Complejidad Técnica, se debe cuantificar con valores de 0 a 5.

La siguiente tabla muestra a los factores, la descripción y el peso.

Factor	Descripción	Peso
E1	Familiaridad con el modelo del proyecto utilizado	1.5
E2	Experiencia en la aplicación	0.5
E3	Experiencia en Teoría Orientado a Objetos	1
E4	Capacidad del Analista Líder	0.5
E5	Motivación de los involucrados en el desarrollo	1
E6	Estabilidad de los requisitos	2
E7	Personal Tiempo Parcial (Part-Time)	-1
E8	Dificultad del Lenguaje de Programación	-1

Tabla 4. Factores que influyen en el Ambiente

- Para los factores E1 al E4, un valor asignado de 0 significa sin experiencia, 3 experiencia media y 5 amplia experiencia (experto).

- Para el factor E5, 0 significa sin motivación para el proyecto, 3 motivación media y 5 alta motivación.
- Para el factor E6, 0 significa requisitos extremadamente inestables, 3 estabilidad media y 5 requisitos estables sin posibilidad de cambios.
- Para el factor E7, 0 significa que no hay personal tiempo parcial (es decir todos son tiempo completo), 3 significa mitad y mitad, y 5 significa que todo el personal es tiempo parcial (nadie es tiempo completo).
- Para el factor E8, 0 significa que el lenguaje de programación es fácil de usar, 3 medio y 5 que el lenguaje es extremadamente difícil.

El Factor Ambiente se debe calcular con la siguiente ecuación:

$$EF = 1.4 - 0.03 * \sum(\text{Peso}_i * \text{Valor Asignado}_i)$$

Ahora se calcula el factor ambiente para el ejemplo que se está desarrollando

Factor	Descripción	Valor Asignado	Comentario
E1	Familiaridad con el Modelo del Proyecto utilizado	3	Está Familiarizado con el Modelo
E2	Experiencia en la aplicación	4	Se ha trabajado desde el inicio de la aplicación
E3	Experiencia en Teoría Orientada a Objetos	5	Se tiene una fuerte preparación
E4	Capacidad del Analista Líder	3	El principiante, pero tiene la especialidad
E5	Motivación de los involucrados en el Desarrollo	5	Se tiene una gran motivación para el desarrollo de la aplicación
E6	Estabilidad de los requisitos	2	Se esperan cambios
E7	Personal Tiempo Parcial (Part-Time)	0	La persona será la encargada de llevar el desarrollo hasta la finalización
E8	Dificultad del Lenguaje de Programación	3	Se usará el lenguaje C#

El valor del Factor Ambiente será:

$$EF = 1.4 - 0.03 (1.5*3 + 0.5*4 + 1*5 + 0.5*3 + 1*5 + 2*2 - 1*0 - 1*3)$$

$$EF = 0.83 \text{ (Factor Ambiente)}$$

Se está listo para calcular los Casos de Uso ajustados. Los dos factores anteriormente calculados ayudaran a ajustar los casos de uso.

$$\mathbf{UCP = UUCP \times TCF \times EF}$$

$$UCP = 92 * 0.945 * 0.83$$

$$UCP = 72.16 \text{ (Puntos de Casos de Uso)}$$

Con este último cálculo se podrá calcular el Esfuerzo.

3.- Esfuerzo.

Un investigador llamado Gustav Karner propuso, en el año 1993, que para cada Punto de Caso de Uso requiere 20 horas-hombre. Pero, en el transcurso del tiempo se ha perfeccionado al siguiente criterio:

- Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.
- Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.
- Si el total es 2 o menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- Si el total es 3 o 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas-hombre.
- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

Por esta teoría, se determina que cada Punto de Caso de Uso tendrá 20 horas-hombre para nuestro ejemplo de desarrollo.

Para calcular el esfuerzo se debe aplicar la siguiente ecuación:

$$E = UCP * CF$$

Donde:

E = el Esfuerzo

CF = Factor de Conversión

UCP = Casos de Uso Ajustados

El Factor de Conversión será 20 horas-hombre, (CF=20 para el ejemplo tomado)

Se debe tomar en cuenta que los cálculos hechos para estimar de esfuerzo solo pertenecen a la parte del ciclo de vida de la implementación o codificación.

Para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software.

Para ello se puede tener en cuenta el siguiente criterio, que estadísticamente se considera aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la siguiente aproximación:

Actividad	Porcentaje (%)
Análisis	10
Diseño	20
Programación	40
Prueba	15
Sobrecarga (otras actividades)	15

Tabla 5. Porcentajes que involucran el tiempo de desarrollo

Estos porcentajes pueden variarse considerando la experiencia que se tenga en el desarrollo y se debe justificar la variación.

Para el ejemplo que se está desarrollando el cálculo del esfuerzo será:

$$E = 72.16 * 20$$

$$E = 1443.204 \text{ Horas-Hombre}$$

Si se toma en cuenta que se trabaja 8 horas diarias y 20 días al mes se tardaría en implementar esta aplicación 9 meses para una sola persona.

Para calcular el total del esfuerzo se debe calcular, por una regla de tres las otras fases.

Actividad	Esfuerzo (E)
Análisis	360.801
Diseño	721.602
Programación	1443.204
Prueba	541.2015
Sobre Carga (Otras Actividades)	541.2015
TOTAL	3608.01

Si se toma que se trabaja 8 horas al día y 20 días al mes (160 horas al mes) el tiempo total para el desarrollo 22 meses. Bastante tiempo para una sola persona.

$$T_{des} = 3608.01 / 160 = 22.5501 \text{ meses.}$$

Pero si se trabaja en equipo, por lo menos cuatro personas que se encarguen de las primeras cuatro actividades, el tiempo de desarrollo seria de 6 meses.

$$T_{des}(\text{equipo}=4) = 22.5501 / 4 = 5.6375 \text{ (6 meses)}$$

Pero si asignamos un número de personas que se ocupen en cada actividad se puede calcular el tiempo de desarrollo por cada actividad tomando el esfuerzo de la actividad.

Actividad	Esfuerzo(E)	No Personas	$T_{des} = E(\text{actividad}) / (\text{horas mes} * \# \text{Personas})$	Meses
Análisis	360.801	2	$360.801 / (160 * 2) = 1.1275$	1
Diseño	721.602	2	$721.602 / (160 * 2) = 2.255$	2
Programación	1443.204	3	$1443.204 / (160 * 3) = 3.0067$	3
Prueba	541.2015	2	$541.2015 / (160 * 2) = 1.6913$	1
Sobre Carga (Otras Actividades)	541.2015	2	$541.2015 / (160 * 2) = 1.6913$	1

Para calcular el costo de desarrollo se toma como parámetro el sueldo promedio un trabajador tomamos como ejemplo de \$10,000.

Esto significa que por hora se paga \$62.5. $(10,000/160)$ [Si multiplicamos el total del esfuerzo por el valor por hora el resultado sería: \\$225,500.625](#)

$$\text{\$hora} = \text{Sueldo} / \text{horas-mes} = 10,000 / 160 = 62.5$$

$$\text{\$Total-proy} = \text{Esf-total} * \text{\$hora} = 3608.01 * 62.5 = 225500.625$$

Podemos calcular el costo por cada actividad tomando el Esfuerzo de cada actividad por el costo de hora. ($\text{\$act} = \text{Eact} * \text{\$hora}$)

Actividad	Esfuerzo(E)	Costo por actividad
Análisis	360.801	$360.801 * 62.5 = 22,550.0625$
Diseño	721.602	$721.602 * 62.5 = 45,100.125$
Programación	1443.204	$1443.204 * 62.5 = 90200.25$

Prueba	541.2015	541.2015*62.5 33,825.0938	=
Sobre Carga (Otras Actividades)	541.2015	541.2015*62.5 33,825.0938	=
TOTAL:	3608.01 total	horas-\$ 225,500.625	

De esta forma se ha culminado la estimación del Esfuerzo, Costo y Tiempo.

E(Esfuerzo-total) = 3608.01 Horas

Tdes-total = 6 meses

\$Costo-total = 225,500.625 pesos

Nuevas estimaciones ahora con diferentes sueldos:

Sueldo:

Analista: 10,000
Diseño: 15,000
Programador: 20,000
Tester: 10,000
Otras: 10,000

Costo Hora por Rol: Sueldo / Total Horas mes

1 Hora Análita: $10,000/160 = 62.5$
1 Hora Diseñor: $15,000/160 = 93.75$
1 Hora Prog: $20,000/160 = 125$
1 Hora Tester $10,000/160 = 62.5$
1 Hora Otras: $10,000/160 = 62.5$

Actividad	Esfuerzo(E)	Costo por actividad
Análisis	360.801	$360.801*62.5 = 22,550.0625$
Diseño	721.602	$721.602*93.7 = 67,614.1074$
Programación	1443.204	$1443.204*125 = 180,400.5$
Prueba	541.2015	$541.2015*62.5 = 33,825.0938$
Sobre Carga (Otras Actividades)	541.2015	$541.2015*62.5 = 33,825.0938$
TOTAL:	3608.01 horas-total	\$ 338,214.8575

$E(\text{Esfuerzo-total}) = 3608.01 \text{ Horas}$

$T_{\text{des-total}} = 6 \text{ meses}$

$\$Costo\text{-total} = 338,214.8575 \text{ pesos}$

Como se puede entender es importante tener una experiencia alta para realizar la estimación del número de transacciones por cada caso de uso, además de definir correctamente los pesos de los factores.

Un detalle que nos lleva a reflexionar es el monto del salario mínimo nacional utilizado en el cálculo anterior.

Si se hace una estimación solo se gana \$25 por día trabajado. Se gana menos que una persona que realiza actividades técnicas, ya que estas personas tienen un pago de \$50 por día.

Siendo una actividad muy compleja el desarrollar software, como se ha descrito en estos documentos, no es posible que se gane por día \$25.

Aunque muchos dirán que es fácil hoy en día, implementar software, pero surgen preguntas... ¿A qué costo? ¿A qué Calidad? ¿Cuánto durará el Software? ¿La persona que ha desarrollado software se convertirá en un “activo fijo” en la organización? Es claro que surgen más preguntas, a las cuales se debe responder con claridad.

Una vez determinado el esfuerzo, tiempo y costo, podemos realizar una planificación de las actividades principales del desarrollo de software.

Este curso no entra en detalles de la planificación de las actividades del Desarrollo de Software.

No se debe olvidar que la planificación es un proceso que va desde la determinación de las actividades pasando por la definición de los riesgos hasta la gestión del proyecto.