



Ingeniería en Desarrollo de software
Semestre 5

Programa de la unidad didáctica:
Métricas de desarrollo de software (PSP)

Unidad 3. Planeación: Recursos y calendario

Clave:
15143529

Ciudad de México, febrero de 2025

Universidad Abierta y a Distancia de México





Índice

Unidad 3. Planeación: recursos y calendario	3
Presentación de la unidad.....	3
Logros.....	3
Competencia específica.....	3
3.1. Estimación de recursos	4
3.1.1. Planeando el recurso	6
3.1.2. Estimando el tiempo de desarrollo	6
3.1.3. Combinación de estimaciones	7
3.2. Estimación de calendario	9
3.2.1. Estimación de cronogramas.....	9
3.2.2. Seguimiento del concepto de valor ganado.....	12
3.2.3. PSP 1.1.....	14
Cierre de la unidad.....	16
Para saber más	16
Fuentes de consulta.....	16



Unidad 3. Planeación: recursos y calendario

Arrancamos ahora la tercera y última unidad de la unidad didáctica: Métricas de desarrollo de software; esta unidad aborda los planes de proyectos que son una lista de actividades que conducen al líder del proyecto hacia el logro del producto esperado, usando y controlando los recursos asignados.

Esto es a nivel proyecto y trabajo en equipo TSP. A nivel personal también puedes planear las actividades necesarias para realizar los programas.

Notarás que el plan contiene la descripción de la actividad y el tiempo esperado de desarrollo, el cual, deberá ser estimado en un inicio y, a medida que vas aplicando en el proceso, notarás que los datos que vas generando te servirán para obtener estimaciones más precisas.

Presentación de la unidad

Hasta este punto ya conoces como estimar tus programas; es por ello que, tomando como base estos conocimientos, te ayudarán a generar planes más precisos para tus programas y proyectos. En otras palabras, podrás generar un calendario como una parte importante de la planeación, del seguimiento a cambios y generación de reportes de gestión. Además, te ayudará a manejar incidentes y desviaciones que pueden controlarse de manera oportuna. Revisa el propósito de la unidad para que no pierdas de vista el alcance y puedas lograrlo sin problema.

Logros

- Identificar los recursos que deben ser planeados y el tiempo de desarrollo por medio de la estimación.
- Analizar aspectos básicos para combinar estimaciones, los elementos que intervienen para estimar un calendario y el concepto de valor ganado.
- Desarrollar un programa aplicando el procedimiento del PSP 1.1

Competencia específica

- Aplicar la estimación de recursos y tiempos para planear el desarrollo de un programa a través de la elaboración de un calendario con el proceso PSP 1.1.



3.1. Estimación de recursos

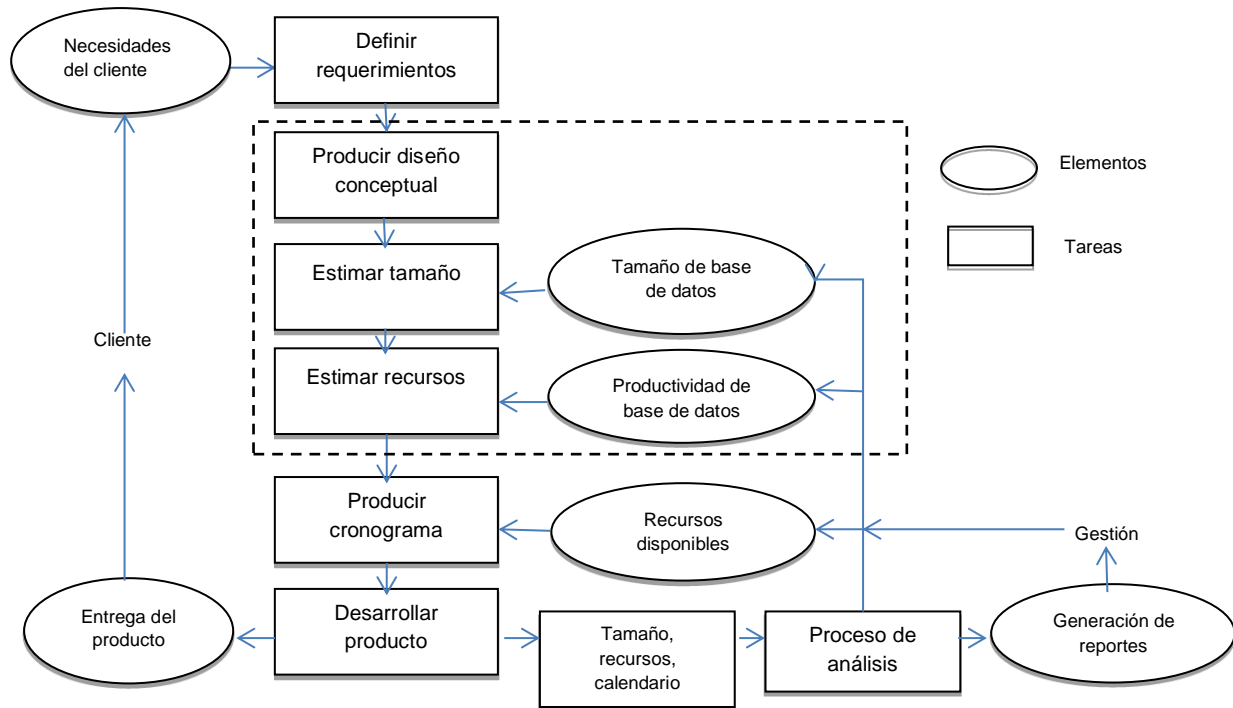
El principal recurso con el que se debe contar son los requerimientos, por lo cual es necesario contar con una adecuada descripción de éstos, ya que de lo contrario no se podría comenzar. Los requerimientos que son completamente válidos para PSP son los que cumplen los siguientes criterios:

1. **Accesibles:** para realizar planes se tiene la necesidad de información y debes contar con el acceso a ésta. Debe estar ordenada, saber dónde se ubica y debe ser consistente y no redundante.
2. **Claros:** lo más importante para PSP es la importancia de la calidad en los datos. Si los datos no están completos o no son lo suficientemente claros no podrán ser utilizados con confianza.
3. **Específicos:** PSP exige que los datos de las plantillas sean llenados correcta y completamente, la ausencia o error en la captura afectará las estimaciones e informes que sean generados.
4. **Precisos:** La precisión es una manera de relacionar la unidad de medida con el total de la magnitud de medición. El nivel de detalle tiene que ver con la precisión. Por ejemplo, para un proyecto con 14 programadores una métrica para administrarlos podría ser horas o días. En PSP se requerirá un nivel de detalle más estricto, por ejemplo, horas y minutos. Manejar segundos ya sería un tiempo demasiado estricto. Para determinar un nivel adecuado de precisión, considera el error introducido por una diferencia de uno en la más pequeña unidad de medida. Por ejemplo, para un proyecto de 5 horas, una unidad de medida de un minuto nos podría introducir errores tolerables, mientras que las unidades de medida de una hora o incluso una décima de una hora pudiera ser demasiado.
5. **Exactos:** todos los puntos son importantes pero la exactitud es crucial. El principal objetivo de hacer planes es producir un plan con una exactitud predecible. Cuando realices tus planes en PSP no debes preocuparte por los pequeños errores que pudieran aparecer al sobre o subestimar actividades, a éstas las llamaremos estimaciones sesgadas. Como tú eres parte de proyectos grandes y eres parte de un equipo, este tipo de errores se balancean con la combinación de valores totales para tener mayor precisión. El método PROBE es una guía para producir planes imparciales y balanceados. (Humphrey, W. 2005, p. 109 - 111).

Para desarrollar el plan, PSP tiene una estructura de trabajo que se muestra en la Figura. Estructura de trabajo para el plan de trabajo de PSP, y que también viste en la Unidad 2. Comienza con las necesidades del cliente, esto ayuda a los programadores o ingenieros de *software* a definir requerimientos. Posteriormente, aplicando el método PROBE (*Proxy Based Estimating*), realizan una estimación y un plan donde cada programador debe realizar un diseño conceptual para poder calcular el tiempo que le llevará hacer el diseño completo. En el siguiente paso, deberán estimar el tamaño del programa y estimar recursos. Posteriormente, ya que conocen el tiempo requerido en cada etapa, estiman el



tiempo que pasarán en el trabajo cada día o semana y de esta manera generan un calendario. Ya que se estableció un calendario, se comienza el desarrollo, lo cual no forma parte de la planeación, pero el registro del tiempo que se genera, sí es importante para realizar estimaciones más precisas de esta etapa. Después de haber completado el trabajo, los programadores o ingenieros de *software* realizan un análisis *postmortem* del trabajo. Actualizan el resumen del plan del proyecto, revisan el desempeño comparando lo real con lo planeado.



Estructura de trabajo para el plan de trabajo de PSP.
(Humphrey, W. 1995, p. 99)

Todo se realiza aplicando el método PROBE para tomar como base la estimación del tamaño del producto. Contabilizan los objetos que se requieren para el diseño conceptual, la cantidad y tipo de métodos de cada objeto, se toma como referencia los estándares de conteo de objetos similares desarrollados previamente. Aplicando una regresión lineal se determina el tamaño para el producto terminado. Para poder aplicar la regresión lineal, como mínimo deben tener un histórico de tres programas anteriores. Ya que estimó el total del trabajo, cada programador usa sus tiempos históricos y ajustan el tiempo necesario para cada fase del trabajo, en la columna **%A la fecha** en el resumen del plan del proyecto.



3.1.1. Planeando el recurso

Otros recursos importantes en la planeación son el tiempo, el personal y, obviamente, *hardware* y *software* necesarios para realizar las actividades. Existe una relación natural entre el tamaño del programa y el tiempo de desarrollo, ya que, entre más grande es el programa, éste involucra mayor tiempo de desarrollo. Dicha relación se puede lograr afectar cuando inyectamos más y mejores recursos al proceso, por ejemplo, establecer un equipo de trabajo sólido formado por varios programadores expertos. En PSP esto comienza con la estimación del tamaño del producto que cada programador desarrolla personalmente, tomando como base sus datos personales y de su productividad; cada programador estima el tiempo para desarrollar el programa. Como vimos en la Figura. Estructura de trabajo para el plan de trabajo de PSP, la estimación del tamaño y del recurso son realizadas por el método PROBE. (Humphrey, W. 1995. Pág. 97-100).

En los siguientes temas veremos cómo estimar el tiempo de desarrollo, la generación de cronogramas de trabajo, estimación de tiempos basándonos en métodos de juicio experto, datos históricos y precisiones estadísticas.

3.1.2. Estimando el tiempo de desarrollo

En PSP cada programador utiliza su información personal para medir su trabajo y estimar mejor los tiempos en nuevos programas. Para lograr esto, registran los datos del tiempo que se gastan en cada fase del proceso, los tamaños de los programas y defectos generados.

El tiempo real registrado deberá contener información como esta: hora en que inició la tarea, tiempo en que se detiene la tarea o cualquier interrupción, por ejemplo, una llamada telefónica, un descanso o alguna conversación no prevista. La idea es que realmente se registre el tiempo invertido en la tarea específica.

El tiempo que gastas en tus proyectos es fluctuante. La primera vez que los miembros de un equipo TSP comienzan a trabajar, generalmente lo hacen con una velocidad entre 12 y 15 horas de trabajo por semana. Después de unas pocas semanas de trabajo ellos pueden a menudo incrementar sus horas de trabajo entre 15 y 17 y ocasionalmente casi alcanzar 20 horas de trabajo. Algunos equipos TSP experimentados pueden alcanzar 20 o más tareas por hora en promedio, a la semana. Lo importante es reunir estos datos sobre el rendimiento personal y utilizarlos para hacer estimaciones y planes más reales y presentar cronogramas más certeros.



Trata de hacer planes realistas que dejen una cantidad razonable por los acontecimientos normales de la vida diaria de trabajo. Esto proporcionará un colchón prudente para los ocasionales momentos de crisis. (Larco, E. 2007. Pág. 30).

En el siguiente tema, volverás a ver técnicas de estimación que serán de utilidad para calcular el tiempo de desarrollo de tus programas o proyectos.

3.1.3. Combinación de estimaciones

Como viste en la Unidad 2, existen muchas técnicas y herramientas de estimación, por ejemplo, el método PROBE que ya fue explicado. En este tema se verán otras técnicas, así como, sus posibles combinaciones. A continuación, se presentan algunas de ellas.

Método Wideband Delphi

Es una técnica basada en juicio experto. Se les pide a varias personas, por separado, que realicen la estimación y la entreguen a quien coordina la dinámica. El coordinador calcula el promedio de la estimación y lo registra. Si algún valor está muy sesgado, realiza una nueva ronda mostrando el promedio e invitando a que se reconsideren los valores reportados. Por ejemplo:

- 1) Primera estimación:
Experto A = 3,000 LOC (Líneas de código)
Experto B = 15,800 LOC
Experto C = 20,800 LOC
- 2) El coordinador calcula el promedio 13,200 LOC y lo proporciona a cada experto, así como las otras estimaciones sin decir de quien son. Los expertos discuten las estimaciones y vuelven a calcular.
- 3) Segunda estimación:
Experto A = 11,200 LOC
Experto B = 13,000 LOC
Experto C = 14,300 LOC
- 4) El coordinador calcula el promedio 12,833 LOC y pregunta a los expertos si están de acuerdo con este promedio. De lo contrario, se realiza un nuevo cálculo.

Ventajas: Los resultados pueden ser muy precisos, se ajusta a las necesidades de la organización y puede aplicarse a cualquier tamaño de proyecto.

Desventajas: consume tiempo de los participantes, si no se maneja adecuadamente los expertos se pueden influenciar y dejar los promedios poco realistas.



Método Analogía

Es otra técnica basada en juicio experto y datos históricos. Debes realizar la separación de los componentes del sistema. Por cada componente, buscar en los datos históricos uno similar y tomar su tamaño como base para la estimación. Por ejemplo: si llevas 2 programas realizados y vas a estimar el tiempo de un tercer programa, lo natural será que revises, si en tus 2 programas anteriores tienes componentes similares y tomes los tamaños registrados de LOC y aplicando tú juicio considerando el alcance del nuevo código realices el ajuste a las líneas de código.

El método de **analogía** lo puedes combinar con el **Delphi**. Cuando no tienen en la empresa valores iniciales, con el método Delphi se pueden generar una serie de valores base, poco a poco conforme vas desarrollando nuevos proyectos, registrarás los valores reales y posteriormente con el método de analogía puedes utilizar los datos históricos.

Método PERT

Este método tiene sus fundamentos en estadística y está basado en los valores de: mínimo esperado, más probable y máximo esperado. Con dichos valores se aplica la siguiente **fórmula**:

a= valor mínimo esperado
b= valor probable
c= valor máximo esperado

$$E = (a + 4b + c) / 6$$

Los valores pueden ser obtenidos a partir de cualquiera de las dos técnicas vistas previamente **Delphi** o **Analogía**.

Por ejemplo:

Valor mínimo esperado (a) = 300
Valor probable (b) = 600
Valor máximo esperado (c) = 1000

$$E = (300 + 4 \cdot 600 + 1000) / 6$$
$$E = 617 \text{ LOC}$$

Si queremos determinar el tiempo para desarrollar el proyecto podemos acudir al modelo de **analogía**, o bien, podemos utilizar la estadística con los datos históricos de productividad.

Tiempo estimado = LOC estimadas del proyecto / media de la productividad.



Media de la productividad = LOC de todos los proyectos realizados / tiempo de todos los proyectos realizados.

Por ejemplo, si tuviéramos los siguientes valores históricos

LOC totales a la fecha = 350

Minutos totales a la fecha = 670

LOC estimadas del proyecto = 230

Media de la productividad = $350/670 = 0.5224$

Tiempo estimado = $230/0.5224 = 440.276$ minutos

Ya que se estimó el tiempo total de desarrollo se debe utilizar el dato para distribuirlo en el plan. A continuación, se explicará cómo aplicar los tiempos estimados en un calendario para generar el plan del desarrollo de tu programa o proyecto. (Mojica, C. & Chapela, L. 2004. pp. 90-95).

3.2. Estimación de calendario

Como hemos visto, podemos estimar el tiempo de esfuerzo requerido para realizar un proyecto. Ya que tienes el tiempo resultante de cualquiera de los estimadores vistos, se tiene que asignar a las diferentes fases del desarrollo, además considerar como se distribuirá semanalmente el trabajo. Todo esto con la finalidad de generar el plan del proyecto.

Como puedes notar, los requerimientos del usuario son necesarios pues en ellos se contiene el alcance del proyecto. Debemos considerar también, otros elementos como: equipo, materiales, *software* y por supuesto los recursos humanos que obviamente son necesarios para el cumplimiento de los objetivos. Esta información será útil para documentar el plan y comenzar a generar el calendario de trabajo.

3.2.1. Estimación de cronogramas

Para generar el calendario de trabajo será necesario contar con la estimación de las horas directas del proyecto, la disponibilidad de los horarios de los recursos y la distribución ordenada de las actividades que se desarrollarán. Posteriormente, necesitarás contar con las estimaciones del tiempo por cada tarea y asignar el tiempo en cada una de las actividades del calendario.

El calendario contendrá todas las actividades que serán necesarias para desarrollar el proyecto con la fecha en que deben iniciar y con la que deben finalizar. Por lo tanto, para crearlo debemos considerar lo siguiente:



1. Contar con el **esfuerzo estimado para el proyecto**.
2. **Disponibilidad de horario**. Se refiere al tiempo que será dedicado para este proyecto; porque, posiblemente, estás involucrado en otras actividades y por lo tanto deberás definir la cantidad de horas que dedicarás diariamente a este proyecto. Por ejemplo:

En un proyecto se tienen la siguiente disponibilidad en los siguientes 4 meses:

- 14 semanas con 30 horas de trabajo por semana = 420 horas disponibles.
- 5 días de vacaciones y 2 días de asueto = 7 días = 42 horas no laborables. 10%
- El tiempo para reuniones, correos y documentación es 5% = 19 horas.
- El tiempo disponible será de $420 - 42 - 19 = 359$ horas.

Saber cuáles son los tiempos que realmente tendremos disponibles es muy importante, ya que con esto podremos asignar las actividades apropiadamente y darnos cuenta si será suficiente el recurso disponible o se necesitará más personal para atender los compromisos.

3. **Orden de las actividades**. El orden de las actividades debe ir de acuerdo con el ciclo de vida de desarrollo que se esté utilizando o cualquier otra estrategia que el equipo de desarrollo haya decidido utilizar.
 - Cada actividad deberá tener un criterio de término
 - Considerar las dependencias entre las actividades: por ejemplo, cuáles comienzan juntas cuáles anteceden a otras y cuáles se preceden.
 - Determinar cuáles actividades son prioritarias y colocarlas en un orden de acuerdo con la naturaleza del proyecto y de los requerimientos del cliente.
 - Asignar costos a los recursos asignados a cada actividad.
 - Determinar el orden de cada actividad.
4. **Define hitos principales en el proyecto**. Al llegar a uno de los principales objetivos o entregables importantes del proyecto, se puede marcar en el calendario la fecha como un hito. Estos no tienen duración, son marcas en calendario que nos indicarán si hemos alcanzado logros.
5. Contar con una **herramienta para cronogramas** con una representación visual como lo es un Gantt (Ruiza, M., Fernández, T., Tamaro, E., Durán, M. 2013. Pág. 1), en el que las barras muestran el esfuerzo requerido en un tiempo determinado. O bien contar con una plantilla estándar para que el manejo del calendario nos de consistencia en el registro y análisis de los datos de todos los proyectos. Por ejemplo, observa el siguiente formato:



Plantilla de Planificación de Tareas (*Task Planning Template*)

Alumno _____ Fecha _____
 Programa _____ Programa # _____
 Instructor _____ Lenguaje _____

Tarea			Plan					Real			
Programa/Parte	Fase	Nombre de la tarea	Horas Tarea	Horas Acumuladas	Valor Planificado (VP)	VP Acumulado	Semana	Horas Tarea	Semana	Valor Ganado (VG)	VG Acumulado

Figura. Ejemplo de formato de planeación de actividades.
(Humphrey, W. 2005, pp.109 - 111)

6. Para llenar el formato deberás:

- Escribir el nombre o Id del programa y la fase de desarrollo.
- En la columna *Nombre de Tarea* escribir las actividades en el orden natural de realización. Deberás describir exactamente cuándo termina la actividad, por ejemplo “casos de uso terminados”, “plan terminado”, “programa compilado”.
- Registrar para cada actividad la cantidad de minutos estimados para su desarrollo.
- Ir acumulando los minutos en la columna de minutos acumulados.
- Adicionalmente podrías agregar una columna para llevar el consecutivo del total de las tareas a realizar.

Valor ganado. - Debemos entender primeramente el objetivo de valor ganado, el cual consiste en establecer un valor para cada actividad, llevar un seguimiento del desarrollo del proyecto comparándolo con lo planeado, aun cuando el plan cambie. Como resultado obtendremos un valor común para cada actividad en forma de porcentaje que representa lo que esa actividad tiene respecto al total de actividades planeadas. No se gana nada por actividades incompletas. Si el proyecto presenta cambios significativos, se requiere hacer una nueva planeación.

Establecimiento de valor planeado. - En el formato de planeación de actividad: sumar las horas del proyecto, calcular el porcentaje de cada actividad respecto al total de horas. Registrar este porcentaje como el valor planeado para la actividad.



Actividades	Horas	Horas acumuladas	Día	Valor planeado	Valor planeado acumulado
A	2	2	1	5.4	5.4
B	5	7	2	13.5	18.9
C	4	11	3	10.8	29.7
D	7	18	4	18.9	48.6
E	3	21	5	8.1	56.7
F	5	26	6	13.5	70.2
G	6	32	7	16.3	86.5
H	3	35	8	8.1	94.6
I	2	37	8	5.4	100

Tabla. Ejemplo de planeación del cronograma

En el formato de calendario: registrar el Valor planeado acumulado para las actividades que sean completadas cada semana. Se calcula el avance al terminar la actividad A de la siguiente manera $100 \times 2/37 = 5.4$, el 2 es del tiempo planeado, el 37 es el total del tiempo que se va a invertir en el proyecto. Y luego en la columna de Valor planeado acumulado recopilar el valor planeado por cada día.

Observa que la columna de Valor planeado es el porcentaje de avance diario que al terminar formará el 100% de avance. Por lo tanto, mantener de manera disciplinada el seguimiento al calendario nos permitirá que, cuando llevemos cierto avance del proyecto, podamos predecir la fecha de término, tomando como referencia la forma en que se ha estado desarrollando las actividades del plan, en el siguiente tema se te explicará cómo hacer este proceso. (Humphrey, W. 1995, pp.145-1155).

A continuación, tendrás la oportunidad de analizar todos estos conceptos en una actividad que te ayudará a entender cómo se aplican e interpretan.

3.2.2. Seguimiento del concepto de valor ganado

Seguimiento al calendario

Ya que se completó el plan, con los tiempos estimados, actividades ordenadas, hitos registrados, costos y fechas de inicio y fin. Nos enfrentamos a la realidad de ponerlo en práctica. Ahora, hay que cuidar que los tiempos se cumplan conforme fueron planeados, que los recursos sigan disponibles y que las fechas se cumplan. Hay que pensar en los posibles riesgos que pudieran retrasar o hasta impedir que las actividades del proyecto se realicen.



El seguimiento del proyecto sería simple si siempre las actividades se completaran en el orden previsto sin que se agreguen nuevas actividades ni que se eliminen. Pero la realidad es muy diferente, esto nunca sucede, los requerimientos siempre cambian, las actividades se eliminan y otras se agregan, la generación de errores es común.

El cumplimiento, o no, de los hitos nos permitirán saber si vamos bien con la estrategia que estamos llevando o debemos realizar cambios o incluso decidir si se sigue o no con el proyecto.

Retomando el concepto de **valor ganado**, debemos registrarlo y la fecha en que fue completada la actividad, para poder dar seguimiento al proyecto. Agregar el valor ganado a la fecha en la columna de valor ganado acumulado.

Formato de calendario: registra el valor ganado acumulado para cada semana a medida que se completan las actividades. Comparar el valor ganado con el valor planeado. Y, como lo vimos anteriormente, una actividad gana valor sólo cuando está terminada. Al completar la actividad, su valor se registra al igual que el valor acumulado hasta ese día. Por lo tanto, el valor ganado nos sirve para saber el estado de nuestro proyecto, es decir, si vamos con retraso, adelantados o conforme al plan.

Proyectar el término del proyecto. - En todos los proyectos se pretende que se cumplan los objetivos de acuerdo con el plan, sin embargo, algunas veces podrán presentar desviaciones ya sea por atrasos o por cambios y en la medida en que vayan avanzando continuará ganando valor. Con la información que se generó anteriormente se determina la fecha de término proyectada, a menos que: la velocidad del proyecto sea cambiada de tal manera que el trabajo restante pueda ser reducido por debajo de lo planeado.

Para llevar el seguimiento al proyecto deberás registrar la fecha o día en que se completaron cada una de las actividades. Recuerda que hay actividades que no se completan en un solo día, por ejemplo, observa en la tabla siguiente, la actividad C, debió terminarse el día 3, sin embargo, fue completada hasta el día 4. Por lo tanto, en la columna de *Valor ganado* no puede tomar el siguiente porcentaje del acumulado planeado, permanece igual. Esto va generando un retraso con respecto a lo que se tenía planeado.



Plan									
ID	Actividades	Horas	Horas acumuladas	Día	Valor planeado	Valor planeado acumulado	Día de término	Valor ganado	Valor ganado proyectado
1	A	2	2	1	5.4	5.4	1	5.4	5.4
2	B	5	7	2	13.5	18.9	2	18.9	18.9
3	C	4	11	3	10.8	29.7	4	18.9	18.9
4	D	7	18	4	18.9	48.6	5	29.7	29.7
5	E	3	21	5	8.1	56.7	6	48.6	48.6
6	F	5	26	6	13.5	70.2			58.32
7	G	6	32	7	16.3	86.5			68.04
8	H	3	35	8	8.1	94.6			77.76
9	I	2	37	9	5.4	100			87.48
10				10					97.2
11				11					100

Tabla. Ejemplo de planeación del cronograma

En el ejemplo anterior, observa la columna *Valor ganado proyectado*, puedes notar que en la actividad 5 del proyecto, se tiene un avance del 48.6%, por lo tanto, en promedio cada día tenemos un avance de 9.72 (48.6/5). En este punto, podemos pronosticar cuándo terminará el proyecto si se continúa trabajando a este ritmo. A partir del día 6 le incrementamos el valor 9.72 y lo continuamos hasta llegar al 100%. Esto nos lleva a terminar el proyecto hasta el día 11 en lugar del día 10 como originalmente estaba planeado.

Poder estimar el término del proyecto, le ayuda al líder a tomar decisiones a tiempo para cubrir sus metas; por ejemplo: agregar un recurso para acelerar el tiempo de desarrollo y evitar que una desviación en el tiempo afecte la fecha de entrega de la etapa. Si se trata del proyecto de un solo programador, éste podrá decidir inyectar más tiempo para evitar que la desviación, no afecte la entrega o bien negociar con el cliente un cambio de fechas, esto tomando en cuenta si la desviación fue ocasionada por el cliente. (Humphrey, W. 2005. Pág. 115).

El plan con valores estimados y su seguimiento forman parte del proceso PSP 1.1. En el siguiente tema se presentan las instrucciones para aplicarlo.

3.2.3. PSP 1.1

Para entender cómo se realiza el proceso PSP 1.1 deberás leer las siguientes instrucciones.

Propósito Guiar a los desarrolladores de programas a nivel módulo.



- | | |
|---------------------------|--|
| Entrada | <ul style="list-style-type: none">○ Descripción del problema○ Formato del resumen del plan del proyecto PSP1.1○ Plantilla de estimación del tamaño○ Los datos históricos de tamaño y tiempo (estimado y real)○ El tiempo y registros de defectos○ Estándares del tipo de defecto, codificación y tamaño○ Cronómetro (opcional) |
| Planeación | <ul style="list-style-type: none">○ Obtención y definición de requerimientos○ Utilización del método PROBE para estimar el tamaño añadido y modificado de este programa, así como el tiempo de desarrollo requerido○ Completar la plantilla de la estimación del tamaño○ Completar una plantilla de Planificación de Tareas○ Completar una plantilla de Planeación Programación○ Introducción de los datos del plan en forma Resumen del Proyecto Plan○ Completar el registro de tiempos |
| Desarrollo | <ul style="list-style-type: none">○ Diseñar el programa○ Aplicar el diseño○ Compilar el programa, corregir y registrar todos los defectos encontrados○ Completar los registros de tiempos |
| Postmortem | Completar el formato del resumen del plan de proyecto con los tiempos reales, defectos y tamaño. |
| Criterios de éxito | <ul style="list-style-type: none">○ Un programa probado a fondo○ Resumen del Plan del proyecto completo con datos reales○ Plantillas llenadas completamente del tamaño, actividades y planificación del calendario completas○ Los formularios de PIP○ - Registro completo del tiempo y defectos. (Alvarado, A. 2008. Pág. 22). |

Este proceso, al igual que los otros, tiene las mismas etapas, pero con nuevos formularios.



Cierre de la unidad

Con esta unidad cerramos el curso de PSP; cómo pudiste darte cuenta, es un proceso de mejora personal, sin embargo, si todos los programadores de la empresa lo utilizaran podrían llevar a un mejor nivel la gestión y forma de trabajo de los proyectos que desarrollen. De manera personal PSP ayuda al programador a controlar, gestionar y mejorar su trabajo por medio del uso de plantillas y procesos que si utiliza adecuadamente servirán para conseguir elementos de trabajo más confiables y predecibles.

Además, revisaste las características de esta disciplina generada por Watts Humphrey, misma que fue el resultado de investigaciones con múltiples proyectos de la industria del *software*. La metodología PSP es parte de un proceso cíclico para ayudar al mejoramiento personal de las actividades y que el primer paso para producir un *software* de calidad es decidir que la calidad es importante, por lo cual se debe establecer el objetivo de producir programas libres de defectos y para tener oportunidad de cumplir esto, debes medir la calidad de tus programas, así como, buscar la mejora de estos.

Con lo anterior, concluimos la unidad didáctica **Métricas de Desarrollo de Software** ¡Felicidades!

Para saber más

Para profundizar en la metodología PSP, te recomendamos que consultes el **Material de apoyo de la Unidad 3**, donde encontrarás lecturas que te apoyarán en tu proceso de aprendizaje.

Fuentes de consulta

Bibliografía básica

Blokdyk, G. (2020). Personal software process: A complete guide. 5STARCooks.

Humphrey, W. (1995). A discipline for software engineering (The complete PSP book). Addison-Wesley.

Humphrey, W. (2005). PSP: A self-improvement process for software engineers. Addison-Wesley.



Zapata, J., García, J., & Cerrada, J. (2001). *Introducción al proceso software personalSM*. Addison-Wesley.

Bibliografía complementaria

Alvarado, A. (2008). *Desarrollo de sistemas con PSP y TSP*. México, D.F.

Larco, E. (2007). *Uso del PSP (Personal Software Process) en el desarrollo de software*. Quito. <http://bibdigital.epn.edu.ec/bitstream/15000/345/1/CD-0760.pdf>

Mojica, C., & Chapela, L. (2004). *Técnicas de programación con calidad*. Yucatán, México.

Ruiz, M., Fernández, T., Tamaro, E., & Durán, M. (2013). *Biografías y vidas*. Barcelona, España. <http://www.biografiasyvidas.com/biografia/g/gantt.htm>