



INSTITUTO POLITÉCNICO NACIONAL



ESCUELA SUPERIOR DE CÓMPUTO

Trabajo: "Investigación de recursos y estimación en Ingeniería de Software"

Asignatura: Ingeniería de Software

Unidad temática: 1.- Fundamentos

Elaborado por: *Corro Mendoza Onasis Alejandro*
García Hernández Brenda Jacqueline
López de la Rosa Elliot Moisés
Luis Luis Ángel Alexis
Zariñan Gómez Emilio

Docente: González Ramírez Marko Alfonso

Fecha: Ciudad de México a 17 de septiembre de 2025

RECURSOS

Según Pressman, el planificador debe considerar tres categorías de recursos antes de iniciar un proyecto [1]:

Recursos humanos: La cantidad de personas requeridas para el desarrollo de un proyecto de software se determina después de hacer una estimación del esfuerzo de desarrollo y seleccionar su rol dentro de la organización y la especialidad que desempeñara cada persona.

- Roles Clave:
 - Desarrolladores: Escriben, prueban y mantienen el código.
 - Ingenieros de calidad: Diseñan y ejecutan pruebas para asegurar la calidad del software.
 - Jefes de proyecto: Planifican, ejecutan y supervisan el proyecto.
 - Diseñadores (UX/UI): Crean la interfaz de usuario y la experiencia del usuario.
 - Arquitectos de software: Toman decisiones de diseño de alto nivel y definen los estándares técnicos.
 - Administradores de sistemas: Gestionan la infraestructura y los despliegues.

Recursos de software reutilizables: bibliotecas, frameworks y componentes que reducen el trabajo de desarrollo, pero incrementan la necesidad de integración y pruebas. Se contemplan los elementos intangibles del sistema, que tendrán como objetivo generar los elementos clave de este con ayuda del hardware.

Estos recursos son utilizados con el fin de utilizar de la forma óptima las herramientas con las que se cuenta para el proceso del sistema a construir. Entre los tipos que más se destacan se contempla:

- Entornos de desarrollo: Ayudan en la construcción a nivel de código del propio sistema desde su base más fundamental, facilitando la tarea con técnicas de desarrollo, entre las que más se destacan se cuenta con Visual Studio Code.

- **Sistemas de control de versiones:** Abarcan las herramientas cuyo propósito es el de permitir un desarrollo controlado del sistema, su importancia se basa en el control de los distintos prototipos del sistema, y la identificación de errores en el desarrollo, uno de los más utilizados actualmente es Git.
- **Software de gestión de proyectos:** Estas herramientas son útiles al momento de controlar el desarrollo del proyecto a un nivel operativo, ayudan a generar diagramas, cronogramas y asignar a cada integrante sus respectivas tareas, usualmente este software es utilizado en equipos de trabajo amplios, como Trello y Asana.
- **Bases de datos:** Sistemas que permiten el control y gestión del conjunto de datos requeridos para el correcto funcionamiento de este tipo de sistemas, abarcan datos informativos y sensibles que son necesarios para la implementación de usuarios, los más utilizados son MySQL y Maria DB.
- **Sistemas operativos:** Utilizados en cada uno de los dispositivos requeridos para todo tipo de proyecto, depende del tipo de proyecto el uso de cada uno de los más importantes (Windows, MacOS y Linux).

Recursos ambientales: hardware, herramientas CASE, entornos de desarrollo, redes y condiciones de trabajo.

Estos recursos influyen directamente en la capacidad de producción del equipo y, por lo tanto, en la calidad y viabilidad del proyecto. Esto se determina por su capacidad, potencia y responsabilidad, es necesario considerar que las capacidades técnicas de estos recursos sean acordes a la magnitud del proyecto a producir, algunos de los que destacan son:

- **Servidores:** Contienen de forma física los datos que se administran en las Bases de Datos, se conecta a estos de forma remota y deben ser capaces de abarcar la cantidad de usuarios y la magnitud del sistema correspondiente. Son necesarios para la producción, desarrollo y pruebas del sistema.

- Equipos de cómputo: Las herramientas utilizadas para el desarrollo del sistema, deben tener la capacidad de soportar los requerimientos operativos del software que es necesario utilizar.
- Dispositivos móviles: Una herramienta versátil que puede ayudar en aspectos como la comunicación, gestión y pruebas en distintas plataformas.
- Infraestructura de red: Dispositivos indispensables para la conexión a base de datos y la comunicación remota entre los integrantes del proyecto, algunas herramientas requieren de una conexión a internet.

Recursos Financieros: Son los tipos de fondos disponibles, incluyendo dinero en efectivo, dinero en efectivo generado por las operaciones y los fondos disponibles de los mercados financieros. Generalmente no se consideran como parte de las operaciones, sin embargo, pueden ser decisivos en habilitar o restringir la aplicación de la estrategia. Es importante recalcar que también se toman en cuenta los salarios del equipo hasta la adquisición de licencias de software y hardware. [8]

- Componentes del presupuesto:
 - Costos de Personal: Salarios y beneficios del equipo.
 - Costos de Infraestructura: Adquisición o alquiler de hardware y servicios en la nube.
 - Licencias de Software: Costos de herramientas, sistemas operativos o librerías propietarias.
 - Costos Indirectos: Espacio de oficina, servicios, etc.
 - Formación y Capacitación: Cursos y certificaciones para el equipo.

El tiempo como recurso: El tiempo es un recurso finito y no recuperable. Su gestión es crítica y se mide a través del cronograma del proyecto.

- **Gestión del Tiempo:**
 - Cronograma del Proyecto: Define las fases, tareas, duraciones e hitos.

- Plazos: Fechas límite para la entrega de hitos o del proyecto final.
- Metodologías: El uso de metodologías ágiles (Scrum, Kanban) buscan optimizar el uso del tiempo a través de ciclos de desarrollo cortos (sprints).

Gestión de recursos

La gestión de recursos se refiere al conjunto de actuaciones orientadas a administrar y utilizar de manera efectiva los recursos disponibles en una organización.

Ventajas de una gestión de recursos optimizada

- Evitar imprevistos. La gestión de recursos permite conocer con que recursos se cuenta de antemano, en vistas a evitar problemas que aparezcan.
- Genera equipos compensados. Una gestión de recursos optimizada permite adecuar los recursos con que se cuenta a las fortalezas y debilidades de los equipos humanos. Evitando escenarios en los que existan cargas de trabajo excesivas o poco realistas.
- Visión global. Determinar y especificar que recursos van a ser necesarios y como se van a asignar permite una visión general en la empresa. Detectando las posibles dependencias o la necesidad de ajustes potenciales.

Técnicas para una mejor gestión de recursos

Asignación: Esta técnica se basa en analizar cuáles son los recursos disponibles a partir de informes específicos para distribuirlos de forma adecuada, ya sea allí donde sean más necesarios o mejor aprovechados. Para ello, hay que realizar una planificación y asignación de los equipos, materiales y presupuestos de forma óptima.

Nivelación: Se trata de descubrir qué recursos están siendo infrautilizados o empleados de forma ineficiente. De este modo, el objetivo es optimizar los recursos sin comprometer la calidad o el cumplimiento de las metas propuestas.

Previsión: Trata de predecir las necesidades futuras en un proyecto para destinar a tiempo los recursos necesarios. Esto es posible gracias al análisis de patrones de

demanda o de los cambios y actualizaciones que se pueden producir en un entorno empresarial. [9]

Aspectos necesarios para la optimización de la gestión de recursos

El uso de la gestión de recursos le permite a una empresa contar con un amplio margen de maniobra al momento de identificar y resolver problemas. De esta forma, al momento de identificar desajustes e incertidumbre sobre el aprovechamiento de los recursos de los que dispone un equipo de trabajo, es importante considerar los siguientes aspectos:

- Definir las prioridades y necesidades de acuerdo con un panorama realista: Se comienza evaluando de forma objetiva las necesidades prioritarias con relación a los recursos con los que se dispone, abarcando personal y capacidades operativas en relación con los tiempos requeridos para satisfacerlas.
- Poner en marcha una planificación con un enfoque estratégico: Generar un plan detallado que especifica hacia donde se destinan los recursos con los que se cuenta y las acciones a realizar para eliminar brechas.
- Valorar el outsourcing: Supone acceder a profesionales especializados y con experiencia en las áreas del proyecto para que abran puertas a la mayor eficiencia y resultados en cortos periodos de tiempo.
- Uso de herramientas tecnológicas para la gestión de recursos: Las plataformas modernas de gestión y colaboración (como Jira, Slack o Asana) son cruciales. Centralizan la comunicación, automatizan el seguimiento de tareas, facilitan la colaboración y se integran con otras herramientas, reduciendo significativamente la carga administrativa y mejorando la sincronización del equipo.

ESTIMACIÓN

Se puede decir que la estimación de proyectos es un proceso complejo que gira en torno a la predicción del tiempo, esfuerzo, duración, recursos, el costo y el alcance necesarios para que un proyecto se considere finalizado. [2]

Principios básicos de la estimación

Incertidumbre inicial: en etapas tempranas la precisión es baja; conforme avanza el ciclo de vida se dispone de más información y la estimación mejora (cono de incertidumbre).

Iteratividad: la estimación no es única; debe revisarse y ajustarse durante todo el proyecto.

Basada en datos: la estimación mejora cuando se apoya en métricas (tamaño, productividad) y datos históricos.

Contextualización: un mismo tamaño de software puede requerir más o menos esfuerzo según el dominio, la calidad requerida, el equipo y las herramientas disponibles.

Tipos de estimaciones

Existen varios tipos de estimación según el momento del ciclo de vida y la información disponible:

- Estimación temprana: cuando aún no se tiene diseño detallado, se usan métricas funcionales como puntos de función [5].

- Estimación detallada: cuando hay arquitectura y especificaciones, se utilizan métricas como SLOC/KSLOC y modelos paramétricos como COCOMO II.

- Estimaciones ágiles: centradas en iteraciones cortas, usando técnicas como Planning Poker, T-shirt sizing y story points para medir esfuerzo relativo.

Dimensionamiento del software

La precisión de una estimación de proyecto de software se basa en algunas cosas:

- 1) El grado en el que se estimó adecuadamente el tamaño del producto que se va a construir.
- 2) La habilidad para traducir la estimación de tamaño en esfuerzo humano, tiempo calendario y dinero (una www.FreeLibros.me 600 PARTE CUATRO ADMINISTRACIÓN DE PROYECTOS DE SOFTWARE función de la disponibilidad de métricas de software confiables de proyectos anteriores).
- 3) El grado en el que el plan del proyecto refleja las habilidades del equipo de software.
- 4) La estabilidad de los requisitos del producto y el entorno que soporta el esfuerzo de ingeniería de software.

En el contexto de la planificación del software, el tamaño se refiere a un resultado cuantificable del proyecto de software. Si se toma un enfoque directo, el tamaño puede medirse en líneas de código (LOC). Si se elige un enfoque indirecto, el tamaño se representa como puntos de función (PF). [1]

Enfoques de dimensionamiento

1. Lógica difusa:
Usa razonamiento aproximado. El planificador clasifica el tipo de aplicación en una escala cualitativa (ej. pequeña, mediana, grande) y luego ajusta el tamaño dentro de ese rango.

2. Puntos de función:

Se estiman las características del dominio de información (entradas, salidas, archivos, consultas) para calcular el tamaño funcional del software.

3. Componentes estándar:

El software se ve como un conjunto de componentes genéricos de su área (módulos, pantallas, reportes, archivos, etc.). El planificador estima cuántos habrá y usa datos históricos para determinar el esfuerzo por cada tipo.

4. Dimensionamiento del cambio:

Se aplica en proyectos que modifican software existente. Se estima la cantidad y tipo de modificaciones: reuso, código agregado, modificado o eliminado.

Estimación basada en problema

Estimaciones LOC y PF

1. Enfoque común

- Ambas técnicas parten de un enunciado del ámbito del software, que luego se descompone en funciones o componentes estimables (funciones, clases, objetos, cambios o procesos de negocio).

2. Uso de métricas de productividad

- Se aplican métricas históricas como LOC/persona-mes o PF/persona-mes.

- Es necesario calcular promedios específicos por dominio de proyecto (según tamaño, área de aplicación, complejidad, etc.) para mayor precisión.

3. Diferencias principales

- LOC: exige una descomposición muy detallada; mientras más granular, más confiable será la estimación.
- PF: se enfoca en las características del dominio de información (entradas, salidas, archivos, consultas, interfaces) y en los 14 factores de ajuste de complejidad.

4. Rangos y valores esperados

- Se recomienda dar un rango de valores (optimista, más probable y pesimista) para cada función o característica.
- El valor esperado se calcula con la fórmula PERT:

$$S = \frac{s_{\text{opt}} + 4s_m + s_{\text{pes}}}{6} \quad [1]$$

Esto pondera más la estimación "más probable" y refleja incertidumbre.

5. Limitaciones y verificación

- Ninguna estimación es totalmente confiable; siempre deben compararse varias técnicas.

- El sentido común y la experiencia siguen siendo esenciales para ajustar los resultados.

Técnicas para manejar la incertidumbre

a) Método PERT (Program Evaluation and Review Technique)

Se utilizan tres estimaciones:

Optimista (O): escenario más rápido posible.

Más probable (M): escenario más realista.

Pesimista (P): escenario más lento o complejo

Permite calcular varianza y desviación estándar, para saber cuán seguro es el rango estimado.

b) Rangos porcentuales

Algunos equipos usan $\pm 20\%$ o $\pm 30\%$ sobre la estimación central según la experiencia histórica.

c) Intervalos de confianza en modelos estadísticos

Si se usan modelos paramétricos o ML, se pueden calcular intervalos de confianza para el esfuerzo, usando desviación estándar o técnicas de bootstrapping.

Esto ayuda a la planificación de recursos y a la negociación con clientes[10].

Factores que afectan la estimación

- Tamaño del software (líneas de código, puntos de función, puntos de historia en ágiles).
- Complejidad técnica (arquitectura, integración con otros sistemas, uso de tecnologías nuevas).
- Experiencia del equipo (productividad varía mucho según habilidades y coordinación).
- Entorno organizacional (procesos, metodologías ágiles o tradicionales, madurez en gestión de proyectos).
- Riesgos externos (cambios en requisitos, presupuesto, disponibilidad de hardware o software) [7].

REFERENCIAS

- [1] R. S. Pressman, *Ingeniería de software: Un enfoque práctico*, 7ª ed. México: McGraw-Hill, 2010, cap. 26
- [2] ISO/IEC/IEEE 15939:2017, *Systems and software engineering — Measurement process*.
- [3] A. Idri, A. Abran, and F. Azzam, "Analogy-based software development effort estimation: A systematic mapping and review," *Information and Software Technology*, vol. 58, pp. 206–230, Feb. 2015.
- [4] ISO/IEC/IEEE 15939:2017, *Systems and software engineering — Measurement process*. IEEE/ISO/IEC, 2017.
- [5] ISO/IEC 19761:2011, *Software engineering — COSMIC Functional Size Measurement*. ISO/IEC, 2011.
- [6] COSMIC, *Measurement Manual, v5.0*. COSMIC Consortium, 2020.
- [7] M. Jørgensen y M. Shepperd, "A systematic review of software development cost estimation studies," *IEEE Transactions on Software Engineering*, vol. 33, no. 1, pp. 33–53, 2007.
- [8] M. Bara. "Recursos en la Gestión de Proyectos: más allá de los recursos humanos y materiales". OBS Business School. Accedido el 17 de septiembre de 2025. [En línea].

Disponible: <https://www.obsbusiness.school/blog/recursos-en-la-gestion-de-proyectos-mas-alla-de-los-recursos-humanos-y-materiales>

[9] Slack. "Cómo mejorar la gestión de recursos | Slack". Slack. Accedido el 17 de septiembre de 2025. [En línea]. Disponible: <https://slack.com/intl/es-es/blog/productivity/gestion-de-recursos>

[10] P. Aroonvatanaporn, "Reducing estimation uncertainty with continuous assessment," Proceedings of the 2010 ACM Conference on Software Engineering, pp. 65–74, 2010.