Planificación del Proyecto

Autor:

Juan Ignacio Odetti

Curso / Asignatura:

Diseño de Sistemas

Fecha:

23/09/2025

Institución / Universidad:

UTN Facultad Regional San Francisco

Introducción

La planificación y gestión eficiente de proyectos constituye un elemento fundamental para garantizar el cumplimiento de objetivos en tiempo y forma. En este contexto, las metodologías PERT (Program Evaluation and Review Technique) y CPM (Critical Path Method) se presentan como herramientas esenciales para la estimación de tiempos y la identificación de las actividades críticas que determinan la duración total del proyecto.

El presente documento tiene como objetivo detallar la planificación de un proyecto de desarrollo de software, incluyendo la estimación de tiempos de cada actividad, la identificación de dependencias y la determinación de la ruta crítica. Para ello, se han registrado las actividades de cada fase e iteración del proyecto, asignando tiempos optimistas, más probables y pesimistas para calcular el tiempo esperado mediante la técnica PERT. Posteriormente, se aplicó el método CPM para determinar los tiempos tempranos y tardíos, así como la holgura de cada actividad, permitiendo identificar las tareas críticas que no deben retrasarse para asegurar la finalización puntual del proyecto.

Descripción General

El Gestor de Gastos es una aplicación que permite al usuario:

- Registrar tanto gastos como ingresos (transacciones).
- Editar y eliminar transacciones registradas.
- Registrar objetivos financieros para un periodo determinado.
- Editar y eliminar objetivos financieros.
- Mostrar el progreso en el cumplimiento de un objetivo financiero.
- Programar posibles transacciones.
- Confirmar la realización de una transacción programada.
- Editar y eliminar transacciones programadas.
- Crear etiquetas personalizadas.
- Mostrar el saldo actual del usuario.

Tecnologías Utilizadas 🛠️

React
JavaScript
HTML / CSS
LocalStorage (Entorno de prueba)
Vite (entorno de desarrollo)
Node.js (backend / API)
PostgrestSQL (base de datos ligera)

Organización del Proyecto

Responsable único: Juan Ignacio Odetti.

Roles asumidos: analista, diseñador, programador, tester y usuario final.

Distribución en Iteraciones (RUP adaptado)

1ª Iteración
Actividades: análisis y prototipo inicial.
Requerimientos abordados:
- Registrar objetivos financieros (alta básica).
- Mostrar progreso en objetivos (en prototipo).

2ª Iteración
Actividades: definición de arquitectura y endpoints backend.
Requerimientos documentados (API / modelo de datos):
- Registrar transacciones (gastos/ingresos).
- Editar y eliminar transacciones.
- Editar y eliminar objetivos financieros.
- Mostrar saldo actual del usuario.
- Crear etiquetas personalizadas.

3ª Iteración
Actividades: implementación backend y frontend, pruebas iniciales.
Requerimientos implementados:
- Registrar / editar / eliminar transacciones.
- Registrar / editar / eliminar objetivos financieros.
- Mostrar progreso en objetivos.
- Crear etiquetas personalizadas.
- Mostrar saldo actual del usuario.

4ª Iteración
Actividades: Implementación de la Base de Datos, pruebas.
Requerimientos afinados:
- Programar transacciones.
- Confirmar transacciones programadas.
- Editar y eliminar transacciones programadas.
- Ajustes finales en transacciones, objetivos, etiquetas y saldo.

Relación con PERT/CPM

Iteración 1 (Inicio): A1–A5 → requerimientos, diagramas, prototipo.
Iteración 2 (Elaboración): B1–B4 → arquitectura, endpoints, base de datos, login.
Iteración 3 (Construcción): C1–C4 → implementación backend, frontend, pruebas.
Iteración 4 (Despliege): D1–D2 → base de datos final, hosting, despliegue.

Gestión de Requisitos

Lista inicial → documento de visión.
Ajustes iterativos → en cada fase, según avance.

Gestión de Riesgos

R1: Falta de tiempo → iteraciones cortas y entregables pequeños.
R2: Cambios en requisitos personales → flexibilidad en la construcción.
R3: Errores de integración backend/frontend → pruebas unitarias e integración en cada endpoint.

Plan de Recursos

Humanos: 1 (desarrollador único).
Materiales: notebook personal, Node.js, React, PostgrestSQL / LocalStorage(prueba), GitHub.
Presupuesto: $0 (uso de herramientas gratuitas).

Plan de Calidad

Pruebas unitarias: en funciones de validación de datos.
Pruebas manuales: de interfaz y de flujo de usuario.
Validaciones clave:
- Montos no negativos.
- Fechas coherentes (inicio < fin).
- No duplicidad de objetivos.

Plan de Configuración y Entregas

Repositorio: GitHub privado.

Cálculos de la Estimación

En este análisis se considera que el proyecto será desarrollado dedicando 3 horas por día, 2 días a la semana, ajustando los tiempos estimados a la disponibilidad del responsable del proyecto. Además, los cálculos utilizados se explican de manera simple, con el fin de facilitar la comprensión de las tablas y gráficas que ilustran las relaciones entre las actividades, los tiempos esperados y la ruta crítica.

Esta planificación permitirá una visión integral del proyecto, facilitando la asignación de recursos, la gestión de riesgos y la toma de decisiones estratégicas durante la ejecución del mismo.

Duración estimada del proyecto: 65.3 días = 32.65 semanas = 8.1625 meses.

1ES – Early Start (Inicio Tempran

● Es el día más temprano en que una actividad puede comenzar, considerando que todas sus predecesoras ya terminaron. ES=0

● Para actividades con predecesoras:

ES=max(EF de todas las predecesoras)

Ejemplo:

● Si la actividad A termina el día 5 y B depende de A, entonces ES de B = 5.

2EF – Early Finish (Fin Tempran

● Es el día más temprano en que la actividad puede terminar.

● Fórmula:

EF=ES+TE;

donde TE es el tiempo esperado de la actividad.

Ejemplo:

● Si B tiene ES = 5 y TE = 3 días, entonces EF = 5 + 3 = 8.

3LS – Late Start (Inicio Tardí ● Es el día más tarde en que la actividad puede comenzar sin retrasar la finalización del proyecto.

● Se calcula en el backward pass (de fin a inicio del proyecto).

Ejemplo:

● Si la actividad B debe terminar antes del día 10 y tiene TE = 3 días, entonces LS = 10 − 3 = 7.

4LF – Late Finish (Fin Tardí

● Es el día más tarde en que la actividad puede terminar sin afectar la fecha de finalización del proyecto.

● Se obtiene en el backward pass.

Ejemplo:

● Siguiendo el ejemplo anterior, si LS = 7 y TE = 3, entonces LF = LS + TE = 7 + 3 = 10.

5Holgura (Slac

● Diferencia entre los tiempos tardíos y tempranos:

Holgura = LS−ES = LF−EF ● Indica cuánto se puede retrasar una actividad sin retrasar el proyecto.

● Si holgura = 0, la actividad pertenece a la ruta crítica.

[VER EXCEL]