

Optimización matemática de la planificación de sprints en equipos Scrum

Denis Alejandro Ordoñez Sevilla

Escuela de Matemática y Ciencias de la Computación
Universidad Nacional Autónoma de Honduras

25 de noviembre de 2025

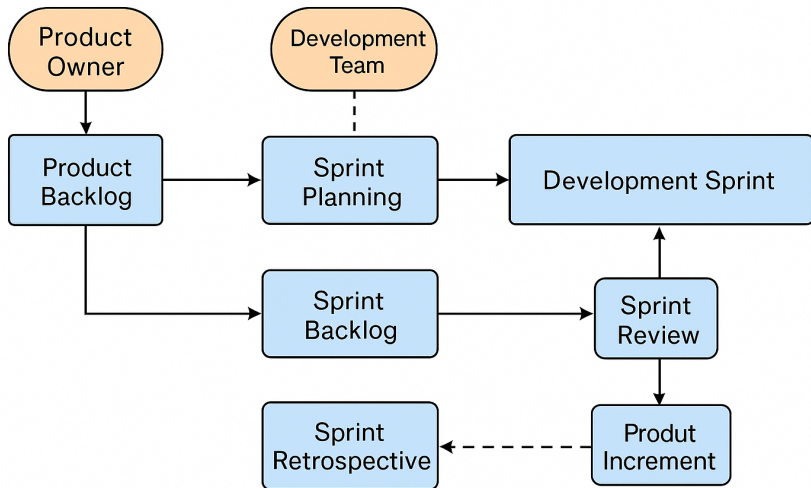
Contenido

- 1 Introducción
- 2 Preliminares
- 3 Definición del problema
- 4 Notación y modelo
- 5 Experimentos y resultados
- 6 Conclusiones y trabajo futuro
- 7 Referencias y Agradecimientos

Motivación

- En entornos ágiles, la asignación manual de tareas suele generar desequilibrios.
- Diferencias en experiencia, habilidades y disponibilidad complican la planificación.
- La optimización matemática permite distribuciones objetivas y eficientes.
- Objetivo: asignar tareas equilibradamente y maximizar el valor del sprint.

Metodología Scrum



Planteamiento

- La planificación de sprints en Scrum debe equilibrar tareas según capacidad, habilidades y prioridades.
- Actualmente se basa en experiencia, lo que puede causar ineficiencias y decisiones subóptimas.
- La complejidad creciente dificulta la asignación óptima mediante intuición.
- Se propone un modelo matemático de optimización para asignar tareas, maximizando valor y equilibrando carga.
- El objetivo es mejorar la eficiencia y calidad en la planificación y ejecución del sprint.
- La planificación empírica no garantiza un equilibrio adecuado ni maximiza el valor entregado.

Conjuntos y parámetros

Conjuntos:

$P = \{p_1, \dots, p_{|P|}\}$ (conjunto de desarrolladores)

$T = \{t_1, \dots, t_{|T|}\}$ (conjunto de tareas)

$W = \{w_1, \dots, w_{|W|}\}$ (conjunto de pesos posibles)

Parámetros:

- $w \in W$ Peso tomado del conjunto de pesos posibles.
- M Constante grande utilizada para la linealización.
- $esSenior_p \in \{0, 1\}$ Indicador de si el desarrollador p es senior (1) o junior (0).
- C_s, C_j Cargas mínimas requeridas para desarrolladores senior y junior, respectivamente.

Variables de decisión y Función objetivo

Variables de decisión:

$x_{p,t} \in \{0, 1\}$	= 1 si la tarea t es asignada a p ,
$y_{t,w} \in \{0, 1\}$	= 1 si la tarea t toma peso w ,
$p_t \geq 0$	peso (puntos) asignado a la tarea t ,
$z_{p,t} \geq 0$	variable auxiliar ($= p_t \cdot x_{p,t}$),
$\text{carga}_p \geq 0$	carga total asignada al desarrollador p .

Función objetivo:

$$\text{máx obj} = \sum_{p \in P} \sum_{t \in T} z_{p,t}.$$

Restricciones (I) – Asignación y pesos

Asignación única de tareas:

$$\sum_{p \in P} x_{p,t} = 1, \quad \forall t \in T$$

Asignación única de peso:

$$\sum_{w \in W} y_{t,w} = 1, \quad \forall t \in T$$

Cálculo del peso de la tarea:

$$p_t = \sum_{w \in W} w y_{t,w}, \quad \forall t \in T$$

Restricciones (II) – Linealización, cargas y límites

Linealización del producto $z_{p,t} = p_t \cdot x_{p,t}$:

$$\begin{cases} z_{p,t} \leq p_t, \\ z_{p,t} \leq Mx_{p,t}, \\ z_{p,t} \geq p_t - M(1 - x_{p,t}), \\ z_{p,t} \geq 0, \end{cases} \quad \forall p, t$$

Carga total asignada a cada desarrollador:

$$\text{carga}_p = \sum_{t \in T} z_{p,t}, \quad \forall p \in P.$$

Cargas mínimas por tipo de desarrollador:

$$\text{carga}_p \geq C_s \cdot \text{esSenior}_p + C_j \cdot (1 - \text{esSenior}_p), \quad \forall p \in P.$$

Rango total del peso asignado:

$$|P_s| \cdot C_s + |P_j| \cdot C_j \leq \sum_{t \in T} p_t \leq 1,2 \cdot (|P_s| \cdot C_s + |P_j| \cdot C_j).$$

Diseño de experimentos

Cuadro: Desarrolladores y cargas mínimas asignadas

Desarrollador	Experiencia	Carga Exp 1	Carga Exp 2
Carlos	Senior	18	27
Sofia	Senior	18	27
Denis	Senior	18	27
Ana	Junior	13	20
Luis	Junior	13	20

- **Experimento 1:** sprint normal, 12 tareas, rango 80–96 puntos.
- **Experimento 2:** sprint extendido , 18 tareas, rango 121–146 puntos.
- **Objetivo:** validar balance, cumplimiento de cargas mínimas y robustez ante cambios de duración.

Diseño de Experimentos

Tareas y Pesos — Exp. 1

Tarea	Peso
t_1	13
t_2	8
t_3	13
t_4	8
t_5	13
t_6	5
t_7	8
t_8	8
t_9	13
t_{10}	3
t_{11}	2
t_{12}	2

Tareas y Pesos — Exp. 2

Tarea	Peso
t_1	13
t_2	2
t_3	13
t_4	5
t_5	8
t_6	13
t_7	13
t_8	2
t_9	3
t_{10}	5
t_{11}	8
t_{12}	13
t_{13}	13
t_{14}	2
t_{15}	3
t_{16}	5
t_{17}	8
t_{18}	13

Resultados: distribución por desarrollador

Experimento 1

Tarea	Carlos	Ana	Luis	Sofía	Denis
t_1	1	0	0	0	0
t_2	0	0	0	1	0
t_3	0	0	0	0	1
t_4	0	1	0	0	0
t_5	0	0	0	0	1
t_6	0	0	1	0	0
t_7	0	1	0	0	0
t_8	0	0	0	1	0
t_9	0	0	1	0	0
t_{10}	1	0	0	0	0
t_{11}	0	0	0	1	0
t_{12}	1	0	0	0	0

Experimento 2

Tarea	Carlos	Ana	Luis	Sofía	Denis
t_1	0	0	1	0	0
t_2	1	0	0	0	0
t_3	0	0	1	0	0
t_4	0	0	0	1	0
t_5	1	0	0	0	0
t_6	0	1	0	0	0
t_7	0	0	0	1	0
t_8	1	0	0	0	0
t_9	0	0	0	1	0
t_{10}	0	1	0	0	0
t_{11}	0	0	0	1	0
t_{12}	0	0	0	0	1
t_{13}	0	0	0	0	1
t_{14}	0	0	0	0	1
t_{15}	1	0	0	0	0
t_{16}	1	0	0	0	0
t_{17}	0	0	0	1	0
t_{18}	1	0	0	0	0

Resultados: Comparación entre experimentos

Comparación entre Experimento 1 y Experimento 2

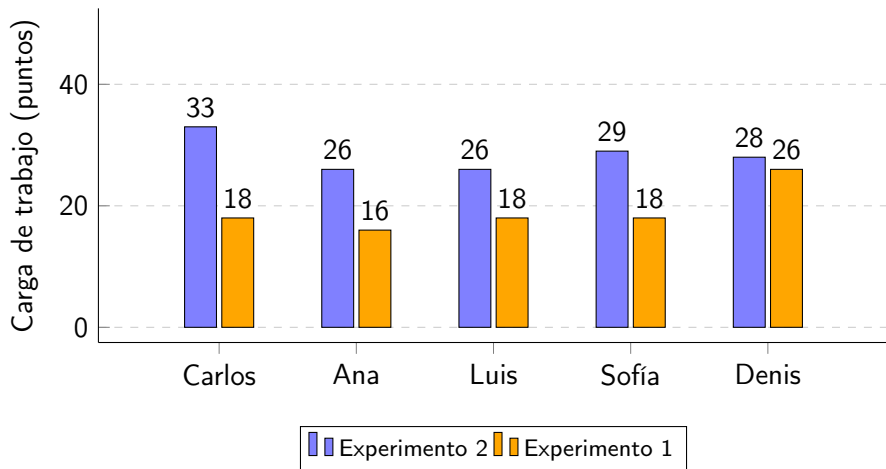


Figura: Comparación de carga de trabajo por persona en dos escenarios.

Conclusiones

- El modelo aseguró la asignación completa y exclusiva de todas las tareas planteadas.
- La distribución de cargas combinó equilibrar el peso de las tareas con la capacidad y especialización de los participantes.
- Se priorizó la factibilidad y eficiencia en la asignación, lo que generó algunas desigualdades en la carga de trabajo.
- Asignar tareas clave a miembros con mayor experiencia mejoró significativamente el valor objetivo y la eficacia global.
- El modelo demostró flexibilidad para adaptarse a distintos escenarios, balanceando equidad y cobertura según la complejidad de las tareas.

Trabajo Futuro

- Añadir criterios como prioridad de tareas, dependencias y urgencia para reflejar mejor condiciones reales.
- Explorar y comparar con otros métodos de optimización, como algoritmos genéticos y heurísticas, para mayor eficiencia.
- Integrar el modelo en plataformas ágiles (Jira, Trello) para facilitar la asignación automática de tareas.
- Evaluar el impacto del modelo en la productividad del equipo, satisfacción de los desarrolladores y cumplimiento de objetivos.
- Desarrollar un sistema de apoyo a la decisión más completo, flexible y adaptable a diferentes equipos y proyectos.

Referencias I

- [1] Burke, R. (2013). *Project Management: Planning and Control Techniques*. John Wiley & Sons.
- [2] Becker, M., Kusters, R., Weijters, T., & ten Berg, B. (2013). Multi-sprint planning and smooth replanning: An optimization model. *Journal of Systems and Software*, 86(6), 1619–1630. doi:10.1016/j.jss.2013.02.010.
- [3] Martínez, D., Pardo, J., & Roldán, C. (2025). Improvements for the Planning Process in the Scrum Method. *Applied Sciences*, 15(1), 202. doi:10.3390/app15010202.
- [4] Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. Scrum.org. Disponible en: <https://scrumguides.org>.
- [5] Cohn, M. (2004). *Agile Estimating and Planning*. Prentice Hall.
- [6] Hillier, F. S., & Lieberman, G. J. (2021). *Introduction to Operations Research* (11th ed.). McGraw-Hill Education.
- [7] Winston, W. L. (2022). *Operations Research: Applications and Algorithms* (5th ed.). Cengage Learning.
- [8] Taha, H. A. (2017). *Operations Research: An Introduction* (10th ed.). Pearson.
- [9] Deniso700. (2025). *Optimización Matemática de la Planificación de Sprints en Equipos Scrum*. Repositorio GitHub: <https://github.com/DENISO700/Optimizacion-matematica-de-la-planificacion-de-sprints-en-equipos-Scrum.git>.

Agradecimientos

¡Gracias por su atención!