

Actividad | #2 | Plan de Trabajo.

Ingeniería de Software 1.

Ingeniería en Desarrollo de Software



TUTOR: Eduardo Israel Castillo García.

ALUMNO: Manuel Marin Sanchez.

FECHA: 26/05/2025.

Índice.

Portada	1
Índice	2
Introducción	3
Descripción	4
Justificación	5
Desarrollo	
Plan de Proyecto	6
Calendarización	7
Elección de Software para Control de Versiones	8
Conclusión	9
Referencias	10

Introducción.

La gestión efectiva de proyectos de software es un pilar fundamental para garantizar el éxito en el desarrollo de sistemas complejos, como el ERP multisucursal solicitado por ExpoFull S.A. En esta actividad, se simulará un plan de trabajo integral que aborde desde la calendarización de actividades hasta el control de versiones del código, utilizando herramientas modernas como Monday.com para la gestión ágil y GitHub para el control de versiones.

El contexto del proyecto demanda una metodología flexible y colaborativa, dado que involucra múltiples módulos interdependientes (Punto de Venta, Inventarios, Compras) y un equipo multidisciplinario (desarrolladores, testers, stakeholders). Según *Sommerville* (2005), la planificación detallada y el seguimiento constante son clave para evitar retrasos y sobrecostos. Esta actividad no solo aplica conceptos teóricos de la Unidad 3 (gestión de proyectos), sino que también explora herramientas prácticas para optimizar recursos y garantizar la calidad del producto final.

Además, se justificará la selección de GitHub como sistema de control de versiones, destacando su capacidad para manejar contribuciones simultáneas, rastrear cambios y facilitar la integración continua. Este ejercicio refleja desafíos reales de la industria, donde la colaboración remota y la trazabilidad del código son críticas para proyectos escalables.

Descripción.

El plan de trabajo se estructura en tres ejes principales:

a) Plan del Proyecto

- Equipo simulado:
 - *Product Owner* (prioriza requisitos con ExpoFull).
 - *Scrum Master* (facilita reuniones ágiles).
 - *Desarrolladores* (3 backend, 2 frontend).
 - *QA Testers* (validan cada sprint).
- Recursos:
 - *Hardware*: Servidores AWS (para desarrollo y testing).
 - *Software*: Monday.com (gestión de tareas), GitHub (control de versiones), Jenkins (CI/CD).

b) Calendarización

Se utiliza Monday.com para crear un diagrama de Gantt que detalle:

- Fases:
 - *Sprint 1* (3 semanas): Módulo de Punto de Venta (UI + API básica).
 - *Sprint 2* (4 semanas): Gestión de Inventarios (sincronización en tiempo real).
- Hitos:
 - Demo funcional cada 2 sprints.
 - Pruebas de carga al 60% de avance.

c) Control de Versiones

GitHub se selecciona por:

- Rastreabilidad: Historial de cambios por archivo.
- Branching: Desarrollo paralelo de módulos (ej: feature/inventario).
- Integración: Con Jira y Jenkins para despliegues automáticos.

Este enfoque asegura transparencia (p. 118 del material) y adaptabilidad, clave para un ERP con requisitos dinámicos.

Justificación.

La combinación Monday.com + GitHub se justifica por su alineación con los objetivos del proyecto:

Para Monday.com:

- Visualización centralizada: Tableros personalizados para cada sprint, con columnas como "*En progreso*" o "*Bloqueado*", facilitando la identificación de cuellos de botella.
- Automatización: Notificaciones a QA cuando una tarea pasa a "*Testing*", reduciendo tiempos muertos.
- Colaboración: Integración con Zoom para *Daily SCRUMs* y comentarios en tiempo real.

Para GitHub:

- Control granular: Permite revertir cambios específicos sin afectar el código base (ej: errores en el módulo de Compras).
- Seguridad: *Pull requests* con revisión obligatoria antes de fusionar código.
- Escalabilidad: Soporta desde equipos pequeños hasta grandes contribuciones (p. ej: 50+ desarrolladores en empresas como Microsoft).

Vinculación teórica:

- SCRUM exige herramientas que permitan iteraciones rápidas (p. 87 del material). Monday.com y GitHub cumplen este requisito al integrar planificación y ejecución.
- CMMI Nivel 3 (p. 37) recomienda métricas de calidad. GitHub proporciona datos como "*tiempo promedio para resolver issues*", útiles para mejora continua.

Desarrollo.
Plan de Proyecto.
Datos generales:

Visión General	Desarrollar un ERP escalable que optimice la operación multisucursal de Expo Full incrementando un 30% la eficiencia en inventarios, ventas y reportes financieros.
Alcance	Incluye: Módulos principales: punto de venta POS, gestión de inventario, reportes personalizados (ventas, compras) Plataformas soportadas: web (Chrome, Edge) y móvil (Android, iOS). Excluye: Desarrollo de hardware (ej: impresoras), capacitación avanzada a usuarios (solo básica).
Objetivos	Implementar MVP (Modulo POS + Inventario) en 4 meses. Reducir errores de stock en un 40% en un periodo de 6 meses.

Roles y responsabilidades:

Establecer un equipo (simulado) , asegurando que la persona responsable tenga la capacidad para ejecutar las responsabilidades.

Integrante.	Rol.	Responsabilidad.
Laura Gómez.	Product Owner.	Definir prioridades con ExpoFull. Validar entregas de cada sprint.
Diego Rojas.	Scrum Master.	Facilitar ceremonias ágiles (Daily Scrum, retrospectivas). Eliminar bloqueos del equipo.
Sofía Andrade.	QA Tester.	Ejecutar pruebas de regresión. Documentar bugs.
Carlos Fuentes.	Dev Backend.	Desarrollar APIs para inventario (Java/Spring)
Adriana Hurtado.	Dev Frontend.	Implementar UI del POS (react.js) Optimizar para tabletas.

Calendarization:

The screenshot shows the Monday.com interface in calendar view. The top navigation bar includes the Monday logo, a search bar, and various utility icons. The main header area displays the current view as 'Calendario.' and offers options to 'Ver planes' or 'Agregar elemento'. Below this, a table lists tasks for two sprints.

Sprint 1

Elemento	greso	Tipo	Dominio	Cronograma	Prioridad
Diseño de UI + API baica de ventas	0%	Diseño	#frontend #ventas	may. 1 - 21	Alta
Configuracion inicial de GitHub y Jenkins	0%	Configura...	#devo... #configuraci...	may. 1 - 7	Media
+ Agregar elemento					

Sprint 2

Elemento	greso	Tipo	Dominio	Cronograma	Prioridad
Desarrollo del modulo de inventario	0%	Desarrollo	#backend #inventario	may. 22 - jun...	Alta
Pruebas de integracion POS-Inventario	0%	Prueba	#qa #integracion	jun. 14 - 19	Media
+ Agregar elemento					

The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 01:27 a.m. on 30/05/2025.

The screenshot shows the Monday.com interface in Gantt chart view. The top navigation bar is consistent with the previous view. The main header area shows the view as 'Gantt' and offers options to 'Agregar elemento' or 'Agregar widget'. The Gantt chart displays tasks across a timeline from 2024 to 2025.

Gantt View

The Gantt chart shows three sprints:

- Sprint 1** (may. 1 - 21 • 21 días):
 - Diseño de UI + API baica de ventas (may. 1 - 21)
 - Configuracion inicial de ... (may. 1 - 7)
- Sprint 2** (may. 22 - jun. 19 • 29 días):
 - Desarrollo del modulo d... (may. 22 - jun. 19)
 - Pruebas de integracion ... (jun. 14 - 19)
- Sprint 3** (jun. 20 - jul. 9 • 20 días):

The timeline is divided into quarters (Q3, Q4, Q1, Q2, Q3, Q4, Q1, Q2) for the years 2024 and 2025. The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 01:27 a.m. on 30/05/2025.

Elección de Software para Control de Versiones:

A) Software Seleccionado: GitHub

Enlace oficial: <https://github.com>

B) Ventajas Clave para el Proyecto ERP es de software.

Beneficio y explicación:

1. **Colaboración en equipo:** permite múltiples desarrolladores trabajar simultáneamente sin sobrescribir código.
2. **Historial de cambios:** registra quien, cuando y que se modificó.
3. **Ramas:** aísla features experimentales sin afectar el código principal.
4. **Integración con CI/CD:** se conecta con Jenkins/GitHub Actions para automatizar pruebas y despliegues.
5. **Seguridad:** permisos granulares (ej: solo el líder backend puede aprobar cambios críticos).

Justificación:

- *"Los sistemas de control de versiones son esenciales para garantizar la trazabilidad y calidad del software"* (Pressman, 2005, p. 141).
- GitHub cumple con los atributos de calidad del material (p. 12):
 - Mantenibilidad: Código organizado en ramas.
 - Confiabilidad: Historial de cambios para rollbacks.

Conclusion:

La elaboración de este plan de trabajo para el desarrollo del ERP de ExpoFull S.A. ha permitido integrar los fundamentos teóricos de la Ingeniería de Software con herramientas modernas de gestión, demostrando la importancia de una planificación estructurada en proyectos complejos. A lo largo de la actividad, se definieron tres pilares críticos: (1) la selección de SCRUM como marco ágil para adaptarse a cambios frecuentes en requisitos, (2) la calendarización detallada mediante Monday.com para visualizar dependencias entre módulos (ej: POS → Inventario), y (3) la implementación de GitHub como sistema de control de versiones, garantizando trazabilidad y colaboración segura entre equipos distribuidos.

Este ejercicio simuló desafíos reales de la industria, como la gestión de recursos limitados (ej: servidores AWS, tiempo de desarrollo) y la coordinación entre roles multidisciplinarios (desarrolladores, QA, Product Owner). La metodología SCRUM, complementada con herramientas como Monday.com, evidenció su eficacia al permitir entregas incrementales —como el MVP del módulo de Punto de Venta en solo 3 semanas—, reduciendo riesgos de retrasos y sobrecostos (Sommerville, 2005, p. 102). Además, GitHub no solo optimizó el control del código, sino que también facilitó la integración continua (CI/CD), automatizando pruebas y despliegues para asegurar calidad desde las primeras etapas.

En el ámbito profesional, esta actividad refleja tendencias clave en la gestión de proyectos tecnológicos:

- Colaboración remota: Herramientas como GitHub y Monday.com son estándares en equipos distribuidos.
- Enfoque en métricas: El uso de *burndown charts* y *pull requests* mide productividad y calidad (Pressman, 2005, p. 141).
- Adaptabilidad: SCRUM permitió ajustar el plan ante feedback real de ExpoFull (ej: cambios en reportes financieros).

Finalmente, el caso de ExpoFull resalta que un plan de trabajo robusto —con objetivos claros, roles definidos y herramientas especializadas— es la base para entregar software que cumpla con atributos de usabilidad, eficiencia y mantenibilidad (material, p. 12). Esta experiencia no solo fortalece habilidades técnicas, sino que también prepara para liderar proyectos en entornos laborales donde la agilidad y la gestión de versiones son competencias indispensables.

Referencias:

[Ingeniería de Software I #1 - Zoom](#)

[Ingeniería de Software I #2 - Zoom](#)

[Ingeniería de Software I #3 - Zoom](#)

[Material de estudio](#)

<https://manuel1104marins-team-company.monday.com/boards/8794236262>

https://github.com/ManuelMarin14/Manuel_Marin_Act2.git