



Actividad | #2 |

Plan de Trabajo

Ingeniería De Software I

Ingeniería en Desarrollo de Software



academiaglobal

TUTOR: Eduardo Israel Castillo García

ALUMNO: Oscar Esteban Sánchez Leyva

FECHA: 26/Mayo/2025

ÍNDICE

ÍNDICE	2
INTRODUCCIÓN	3
DESCRIPCIÓN	4
JUSTIFICACIÓN	5
DESARROLLO	6
PLAN DE PROYECTO.....	6
CALENDARIZACIÓN	9
ELECCIÓN DE SOFTWARE PARA CONTROL DE VERSIONES	12
CONCLUSIÓN.....	13
REFERENCIAS.....	14

INTRODUCCIÓN

En el presente documento se hablará acerca de un plan de trabajo o plan de proyecto, que es un documento fundamental en la gestión de proyectos de software, que detalla las actividades, tareas, recursos y cronograma necesarios para alcanzar los objetivos del proyecto. Se utiliza para asegurar que el proyecto se desarrolle de manera eficiente y se cumpla con los plazos, presupuestos y requisitos definidos.

Elementos clave de un plan de trabajo:

- **Objetivos del proyecto:** Define claramente lo que se quiere lograr con el proyecto, tanto a nivel general como específico.
- **Alcance del proyecto:** Describe lo que se incluye y lo que se excluye del proyecto, para evitar ambigüedades y confusiones.
- **Cronograma:** Establece las fechas de inicio y finalización de cada tarea y actividad, así como los plazos intermedios.
- **Recursos necesarios:** Identifica los recursos humanos, materiales, tecnológicos y financieros que se requieren para llevar a cabo el proyecto.
- **Presupuesto:** Define los costos asociados a cada recurso y actividad, para controlar el presupuesto del proyecto.
- **Riesgos y dependencias:** Identifica los posibles riesgos que podrían afectar al proyecto y las dependencias entre las diferentes tareas y actividades.
- **Responsabilidades:** Define quién es responsable de cada tarea y actividad, para asegurar que se cumpla con los plazos y requisitos.

DESCRIPCIÓN

En una herramienta de administración de proyectos de software o plan de trabajo que detalla todos los aspectos de un proyecto para asegurar su éxito. Define metas, objetivos, alcance, cronograma, riesgos y dependencias, sirviendo como guía para la ejecución del proyecto.

Plan de trabajo en software de gestión de proyectos:

- **Objetivos y metas:** Define qué se espera lograr con el proyecto.
- **Alcance:** Especifica qué tareas se incluirán y cuáles no, para evitar ambigüedades.
- **Cronograma:** Establece las fechas de inicio y finalización de cada tarea y las dependencias entre ellas.
- **Riesgos:** Identifica y evalúa los riesgos potenciales que podrían afectar el proyecto y planifica estrategias para mitigarlos.
- **Dependencias:** Define las relaciones entre las tareas y cómo una tarea afecta a otra.
- **Presupuesto:** Define el costo total del proyecto y cómo se distribuirán los recursos.
- **Tareas y subtareas:** Desglosa el proyecto en tareas más pequeñas y manejables.
- **Asignación de responsabilidades:** Define quién es responsable de cada tarea.
- **Indicadores de progreso:** Define cómo se medirá el progreso del proyecto.
- **Comunicaciones:** Define cómo se comunicará el progreso del proyecto a las partes interesadas.

JUSTIFICACIÓN

La justificación de un plan de trabajo dentro de un software de gestión de proyectos implica explicar las razones por las que dicho plan es necesario y valioso para el proyecto. Se trata de demostrar cómo el plan contribuirá a la consecución de los objetivos, la eficiencia del equipo y la reducción de riesgos.

Elementos claves justificando el plan de trabajo:

Beneficios esperados:

- ¿Cómo el plan de trabajo mejorará el rendimiento del proyecto, la eficiencia del equipo, la comunicación o el control de calidad?

Problemas que aborda:

- ¿Qué deficiencias en la gestión del proyecto el plan de trabajo ayudará a solucionar? ¿Cómo evitará retrasos o errores?

Impacto en la empresa:

- ¿Cómo contribuirá el plan de trabajo a los objetivos estratégicos de la empresa?

Justificación de los recursos:

- ¿Por qué se necesitan los recursos especificados en el plan?

Riesgos mitigados:

- ¿Cómo el plan de trabajo ayudará a identificar y gestionar los riesgos del proyecto?

La justificación del plan de trabajo en un software de gestión de proyectos debe ser clara, concisa y demostrar cómo el plan contribuirá al éxito del proyecto y al logro de los objetivos de la empresa.

DESARROLLO

PLAN DE PROYECTO

Para el desarrollo del ERP de ExpoFull S.A., considerando los módulos mencionados y la necesidad de simular la gestión del proyecto y el control de versiones, el método de desarrollo ágil, específicamente Scrum, sería el más adecuado.

Visión General	El plan del proyecto bajo Scrum se centra en la adaptabilidad y la entrega de valor incremental. No se define un plan rígido al inicio, sino que se construye y refina a lo largo del desarrollo.
Alcance	Crear una lista priorizada de todas las funcionalidades y requisitos del ERP. Los módulos mencionados serían los elementos principales, desglosados en funcionalidades más pequeñas.
Objetivo	Desarrollar un Sistema de Información y Planificación de Recursos Empresariales (ERP) para ExpoFull S.A. que optimice la gestión de ventas, compras, inventario, clientes y empleados en sus más de 50 puntos de venta a nivel nacional.

REQUISITOS Y PLANIFICACIÓN

- 1. Análisis de requisitos:** Recopilar información sobre las necesidades específicas de ExpoFull S.A. y definir los requisitos funcionales y no funcionales del sistema.
- 2. Definir el Product Backlog:** Lista priorizada de todas las funcionalidades, características, mejoras y correcciones que se necesitan para el ERP. Cada elemento debe ser claro y con estimaciones de esfuerzo.
- 3. Sprint Backlog:** Subconjunto de elementos del Product Backlog seleccionados por el Development Team para ser completados en un sprint específico.

4. Asignar roles:

- **Product Owner (ExpoFull S.A. / ITPower):** Representante del cliente, con profundo conocimiento del negocio, responsable de definir y priorizar los elementos del Product Backlog (lista de funcionalidades), maximizando el valor del producto.
- **Scrum Master (ITPower):** Facilitador del equipo, responsable de asegurar que se sigan las prácticas de Scrum, eliminar impedimentos y proteger al equipo de interrupciones externas.
- **Development Team (ITPower):** El equipo multidisciplinario encargado de desarrollar el software, autoorganizado y responsable de la entrega de incrementos de software funcionales.

DESARROLLO

1. **Sprint Planning:** Planificar cada sprint (duración recomendada: 2-4 semanas) seleccionando las tareas del Product Backlog que se abordarán.
2. **Desarrollo:** El equipo trabaja en las tareas planificadas, realizando daily stand-ups para seguimiento y resolución de impedimentos.
3. **Revisión y retrospectiva:** Al final de cada sprint, revisar lo completado y realizar una retrospectiva para identificar mejoras.

Plan de Trabajo ▾

Tabla principal ... Calendario +

Agregar tarea ▾ 🔍 Buscar 👤 Persona ⚙ Filtrar ▾ ⬆️ Ordenar 🗒 Ocultar 📁 Agrupar por ...

▾ Pendientes

<input type="checkbox"/>	Tarea		Responsable	Estado ⓘ	Vencimiento ⓘ		Prioridad	Notas
<input type="checkbox"/>	Pantalla de Acceso	🗨	OE VG JA	Listo	✓	may-28	Alta	Sprint 1
<input type="checkbox"/>	Punto de Venta	🗨	VG	En curso	🕒	may. 31	Baja	Sprint 2
<input type="checkbox"/>	Módulo de Compras	🗨	JA OE	En curso	○	jun. 3	Media	Sprint 3
<input type="checkbox"/>	Registro de Clientes	🗨	OE	Listo	✓	jun-6	Alta	Sprint 4
<input type="checkbox"/>	Registro de Empleados	🗨	VG JA	Detenido	○	jun. 12	Baja	Sprint 5
<input type="checkbox"/>	Generación de Reportes	🗨	JA	Listo	✓	jun-16	Alta	Sprint 6
<input type="checkbox"/>	Gestión de Inventarios	🗨	OE VG	Detenido	○	jun. 22	Baja	Sprint 7

Notas	Texto
Sprint 1	Autenticación de usuarios, roles de usuario.
Sprint 2	Registro de ventas, cálculo de precios, gestión de devoluciones, métodos de pago.
Sprint 3	Creación de órdenes de compra, recepción de mercancía, gestión de proveedores.
Sprint 4	Creación/edición de clientes, historial de compras.
Sprint 5	Gestión de datos de empleados, roles y permisos.
Sprint 6	Reportes de ventas por período, reportes de compras por proveedor, reportes de inventario actual.
Sprint 7	Actualización de stock, alertas de bajo inventario, transferencias entre puntos de venta.

Welcome to monday.com / a new way of working. (s. f.).

<https://oscarestebansanchezleyva13s-team.monday.com/boards/9234797706/>

CALENDARIZACIÓN

La calendarización en Scrum no es un cronograma fijo y detallado a largo plazo, sino un plan adaptativo basado en sprints de duración fija; esto permite entregas frecuentes y retroalimentación constante.

- **Duración de sprints:** Se recomienda iniciar con sprints de 2 a 3 semanas.
- **Planificación del Sprint (Sprint Planning):** Al inicio de cada Sprint, el equipo selecciona los elementos de mayor prioridad del Product Backlog que pueden completar en ese Sprint. Se desglosan en tareas más reducidas (Sprint Backlog) y se estima el esfuerzo.
- **Reuniones diarias (Daily Scrum):** El equipo se reúne diariamente para sincronizarse, revisar el progreso y planificar el trabajo del día.
- **Revisión del Sprint (Sprint Review):** Al final de cada Sprint, el equipo demuestra el incremento de software funcional a los stakeholders (ExpoFull S.A.) para obtener retroalimentación.
- **Retrospectiva del Sprint (Sprint Retrospective):** Después del Sprint Review, el equipo reflexiona sobre el Sprint que acaba de terminar e identifica mejoras en el proceso de desarrollo.
- **Liberaciones (Releases):** La frecuencia de las liberaciones a producción dependerá de la complejidad de los módulos y la capacidad de ExpoFull S.A. para adoptar los cambios. Podrían ser liberaciones mensuales o trimestrales de conjuntos de funcionalidades.

Ejemplo de calendarización inicial (hipotéticamente):

- **Sprint 1:** Pantalla de Acceso (autenticación de usuarios, roles de usuario).
- **Sprint 2:** Punto de Venta (registro de ventas, cálculo de precios, gestión de devoluciones, métodos de pago).
- **Sprint 3:** Módulo de Compras (creación de órdenes de compra, recepción de mercancía, gestión de proveedores).
- **Sprint 4:** Registro de Clientes (creación/edición de clientes, historial de compras).
- **Sprint 5:** Registro de Empleados (gestión de datos de empleados, roles y permisos).
- **Sprint 6:** Generación de Reportes (reportes de ventas por período, reportes de compras por proveedor, reportes de inventario actual).
- **Sprint 7:** Gestión de Inventarios (actualización de stock, alertas de bajo inventario, transferencias entre puntos de venta).

Plan de Trabajo ▾

Integ

Tabla principal Calendario ... +

Agregar tarea ▾

+ Agregar widget

🔍 Buscar

👤 Persona

🔼 Filtrar ▾

Calendario 🔼

lun.	mar.	mié.	jue.	vie.
26	27	28	29	30
		Pantalla de Acceso Osc...		
02	03	04	05	06
	Módulo de Compras Ja...			Registro de Clientes Os...
09	10	11	12	13
			Registro de Empleados ...	
16	17	18	19	20
Generación de Reportes ...				
23	24	25	26	27

Welcome to monday.com / a new way of working. (s. f.-b).

<https://oscarestebansanchezleyva13s-team.monday.com/boards/9234797706/views/196428467>

ELECCIÓN DE SOFTWARE PARA CONTROL DE VERSIONES

Un software adecuado para el control de versiones en este proyecto es Git, con una plataforma de gestión como GitHub o GitLab. La elección se justifica por las siguientes razones:

1. **Distribuido:** Git permite trabajar de manera distribuida, lo que facilita la colaboración entre los miembros del equipo.
2. **Flexibilidad:** Git soporta diferentes flujos de trabajo y puede adaptarse a las necesidades del proyecto.
3. **Seguridad:** Con GitHub o GitLab, se pueden implementar medidas de seguridad como autenticación de dos factores y permisos de acceso.
4. **Integración:** Ambas plataformas ofrecen integración con herramientas de CI/CD, lo que facilita la automatización de pruebas y despliegue.
5. **Comunidad y Soporte:** Git y las plataformas asociadas tienen una gran comunidad y soporte extenso, lo que reduce la curva de aprendizaje y facilita la resolución de problemas.
6. **Plataformas de Hosting (GitHub/GitLab):**
 - **GitHub:** Es la plataforma más grande para el control de versiones, ofreciendo una excelente interfaz gráfica, funciones de colaboración.
 - **GitLab:** Ofrece una suite de DevOps más completa, incluyendo CI/CD integrado, registro de contenedores y más herramientas de gestión de proyectos, todo dentro de la misma plataforma.

La combinación de Scrum para la gestión del proyecto y Git (GitHub/GitLab) para el control de versiones proporcionará a ITPower la agilidad, flexibilidad y robustez necesarias para desarrollar el ERP de ExpoFull S.A. de manera eficiente y colaborativa.

CONCLUSIÓN

En conclusión, un plan de trabajo dentro de una herramienta de gestión de proyectos (software), se resume lo aprendido, se evalúan los resultados obtenidos y se identifican áreas de mejora para futuros proyectos. También se deben mencionar los principales logros, los desafíos enfrentados y las lecciones aprendidas, tomando como referencias los siguientes puntos:

1. **Resumen de los resultados:** Se debe describir de manera concisa los objetivos del proyecto, los resultados alcanzados y si se lograron o no.
2. **Evaluación del proceso:** Se debe analizar cómo se llevó a cabo el proyecto, qué funcionó bien y qué se podría mejorar en el futuro.
3. **Lecciones aprendidas:** Se deben identificar los conocimientos adquiridos durante el proceso, las dificultades encontradas y las estrategias utilizadas para superarlas.
4. **Recomendaciones:** Se deben ofrecer sugerencias para la optimización de futuros proyectos, basándose en la experiencia obtenida.
5. **Impacto del software de gestión de proyectos:** Se debe evaluar el impacto del software en el desarrollo del proyecto, destacando sus beneficios y posibles áreas de mejora.

REFERENCIAS

- Martins, J. (2025, 20 mayo). Plan de proyecto: Cómo crear uno {ejemplos + plantilla} [2025] • Asana. Asana. <https://asana.com/es/resources/project-management-plan>
- Martins, J. (2025a, febrero 15). Scrum: conceptos clave y cómo se aplica en la gestión de proyectos [2025] • Asana. Asana. <https://asana.com/es/resources/what-is-scrum>
- Gurnov, A. (2022, 18 agosto). *Fundamentos de la gestión de proyectos / Guía de gestión de proyectos*. Wrike. <https://www.wrike.com/es/project-management-guide/fundamentos-de-la-gestion-de-proyectos/>
- Prieto, E. (s. f.). *¿Qué es un software para la administración de proyectos? - Southern New Hampshire University*. SNHU. <https://es.snhu.edu/blog/que-es-un-software-para-administrar-proyectos>
- Ermete.Salvatore. (2024, 12 enero). *Gestión de proyectos de la A a la Z: definición, herramientas y software*. CyberPlan. <https://cyberplan.it/es/project-management-dalla-a-alla-z-definicion-herramientas-y-software/>