****

**Actividad 2. Plan de Trabajo**

**Ingeniería de Software 1**

**Ingeniería en Desarrollo de Software**

**Tutor: Eduardo Israel Castillo García**

**Alumno: Gibrán Alfredo Zazueta Fabela**

**Fecha: 18/09/2023**

**Índice**

● Introducción

● Descripción

● Justificación

● Desarrollo

* Plan de proyecto
* Calendarización

● Conclusión

**Introducción**

En este trabajo, se abordará la planificación y ejecución de un proyecto, así como la calendarización de cada una de las actividades. Además, se seleccionará un método adecuado para el control de versiones de software.

La planificación y ejecución de un proyecto son elementos cruciales. Un plan de proyecto bien estructurado permite establecer los objetivos, identificar las tareas necesarias, asignar recursos y definir los plazos. Por otra parte, la calendarización de las actividades es esencial para garantizar un flujo de trabajo eficiente y organizado. Al asignar fechas específicas a cada tarea, se puede visualizar el progreso del proyecto y anticipar posibles retrasos o desviaciones.

En el caso del control de versiones de software, es fundamental para gestionar los cambios realizados en el código fuente de un proyecto. Permite mantener un historial de las modificaciones, facilitando la colaboración entre los miembros del equipo y asegurando la integridad del software. La elección de un método adecuado de control de versiones es crucial para garantizar una gestión eficiente y segura de los cambios.

**Descripción**

El presente trabajo se centra en el desarrollo de un plan de proyecto, la calendarización de las actividades y la elección de un método para el control de versiones de software. Estas etapas son esenciales para garantizar la correcta ejecución y gestión de un proyecto de desarrollo de software.

En primer lugar, se elaborará un plan de proyecto detallado que incluirá el nombre de los participantes, su rol y una descripción de sus responsabilidades. Además de una descripción de la visión general del proyecto, el alcance y los objetivos.

Posteriormente, se procederá a la calendarización de las actividades. Esto implica asignar fechas específicas a cada tarea y establecer los plazos de entrega. La calendarización permitirá visualizar el progreso del proyecto, identificar posibles retrasos y gestionar eficientemente los recursos disponibles. Además, brindará la oportunidad de realizar ajustes y modificaciones en caso de ser necesario.

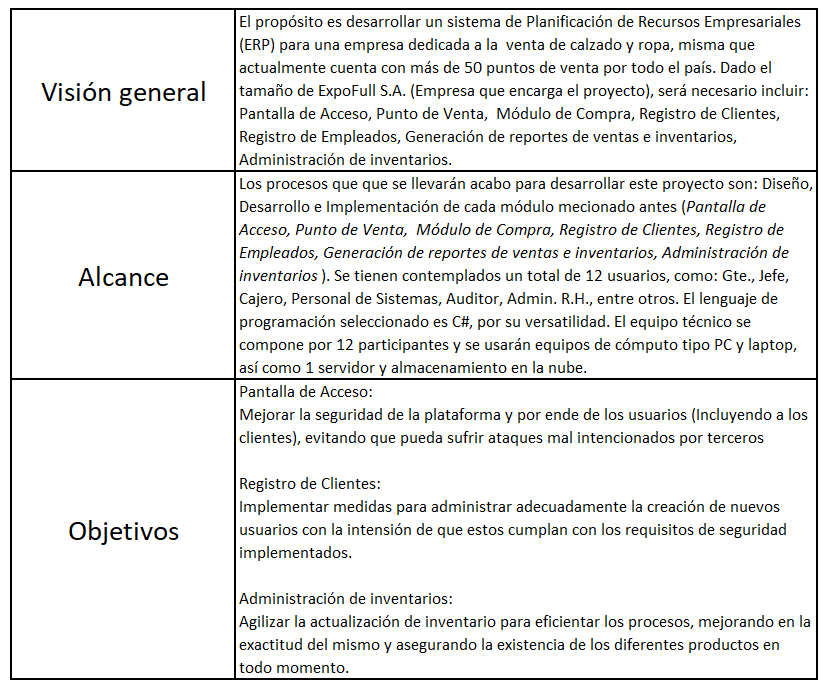
Por último, se seleccionará un método de control de versiones de software adecuado para el proyecto. Para esto, existen diversas opciones disponibles, como Github, Subversion, VisualStudio Code o Mercurial, entre otros. La elección de un método apropiado dependerá de las necesidades y características del proyecto, así como de las preferencias del equipo de desarrollo. El control de versiones permitirá gestionar los cambios realizados en el código fuente, mantener un historial de las modificaciones y facilitar la colaboración entre los miembros del equipo.

**Justificación**

Puesto que el desarrollo de software es una disciplina compleja y que requiere de mucha colaboración, es necesaria una cuidadosa planificación y gestión para garantizar el éxito de los proyectos. En este sentido, la elaboración de un plan de proyecto detallado, la calendarización de las actividades y la elección de un método adecuado para el control de versiones de software, son etapas fundamentales que justifican su inclusión en cualquier proyecto de desarrollo de software. A continuación, se presentan las razones por las cuales estas etapas son esenciales:

1. Elaboración de un plan de proyecto detallado: Esta permite establecer una hoja de ruta clara y definida para el desarrollo del software. Al tener un plan detallado, el equipo de desarrollo podrá comprender claramente las metas y los pasos a seguir, lo que facilitará la coordinación y la toma de decisiones a lo largo del proyecto.
2. Calendarización de las actividades: Implica asignar fechas específicas a cada tarea y establecer los plazos de entrega. Esta etapa permite visualizar el progreso del proyecto, identificar posibles retrasos y gestionar eficientemente los recursos disponibles. La calendarización también brinda la oportunidad de realizar ajustes y modificaciones en caso de ser necesario.
3. Selección de un método de control de versiones de software: El control de versiones es crucial en el desarrollo de software, ya que permite gestionar los cambios realizados en el código fuente, mantener un historial de las modificaciones y facilitar la colaboración entre los miembros del equipo. La elección de un método adecuado de control de versiones garantiza la integridad y la trazabilidad del código fuente, evitando conflictos y errores en el proceso de desarrollo.

**Desarrollo**

**Plan del proyecto:**

Pantalla de Acceso:

* Requerimientos de Hardware: Servidor proxy IBM o Firewall
* Requerimientos de Software: Linux OS, interface de inicio de sesión con métodos de autenticación, base de datos.
* Informes necesarios: Confirmación de inicio de sesión o de rechazo.
* Tiempo estimado de trabajo: 14 días
* Roles involucrados: Experto en Ventas, Analista, Diseñador(es) de interfases,

Desarrollador(es) de software, Tester, Validador.

Punto de Venta:

* Requerimientos de Hardware: POS (**P**oint **O**f **S**ale), Escáner laser, Cajón de dinero, Impresora y PC (Con sus periféricos), Lector de tarjetas, servidor, cables de red.
* Requerimientos de Software: Software para punto de venta que se integre al POS (Con SO Android).
* Informes necesarios: Info. del artículo cargado (Nombre, precio, promoción (si aplica).
* Tiempo estimado de trabajo: 30 días
* Roles involucrados: Diseñador(es) de interfases, Desarrollador(es) de software,

Especialista en seguridad, Experto en Hardware, Experto en ventas y Tester.

Módulo de Compras:

* Requerimientos de Hardware: PC o Laptop (Sin requerimientos especializados).
* Requerimientos de Software: Gestión de proveedores, Solicitud de cotizaciones, Gestión de órdenes de compra, Integración con inventario.
* Informes necesarios: No hay informes específicos.
* Tiempo estimado de trabajo: 20 días
* Roles involucrados: Experto en Ventas, Analista, Diseñador(es) de interfases,

Desarrollador(es) de software, Tester, Validador.

Registro de Clientes:

* Requerimientos de Hardware: PC, periféricos, dispositivos móviles, servidor, cámaras, dispositivos biométricos e impresoras.
* Requerimientos de Software: Software para capturar y validar datos que implemente medidas de seguridad y privacidad, permita gestionar base de datos y que se integre con el resto de sistemas.
* Informes necesarios: Confirmación de alta y/o actualización de usuarios.
* Tiempo estimado de trabajo: 20 días
* Roles involucrados: Experto en Ventas, Analista, Diseñador(es) de interfases,

Desarrollador(es) de software, Tester, Validador.

Registro de Empleados:

* Requerimientos de Hardware: PC, periféricos, dispositivos móviles, servidor, cámaras, dispositivos biométricos e impresoras.
* Requerimientos de Software: Software para capturar y validar datos que implemente medidas de seguridad y privacidad, permita gestionar base de datos y que se integre con el resto de sistemas.
* Informes necesarios: Confirmación de alta y/o actualización de usuarios.
* Tiempo estimado de trabajo: 14 días
* Roles involucrados: Experto en Ventas, Analista, Diseñador(es) de interfases,

Desarrollador(es) de software, Tester, Validador.

Generación de reportes de ventas e inventarios:

* Requerimientos de Hardware: Servidor, PC, Impresora, Escáner (De barras o de ondas RFID) y Redes.
* Requerimientos de Software: Software para Facturación, Resumen de ventas

(Configurable para filtrar por fechas, periodos, productos, por POS o precio).

* Informes necesarios: Totales (Compra, venta y existencia vs. excedente y/o faltante).
* Tiempo estimado de trabajo: 14 días
* Roles involucrados: Experto en Ventas, Analista, Diseñador(es) de interfases,

Desarrollador(es) de software, Tester, Validador.

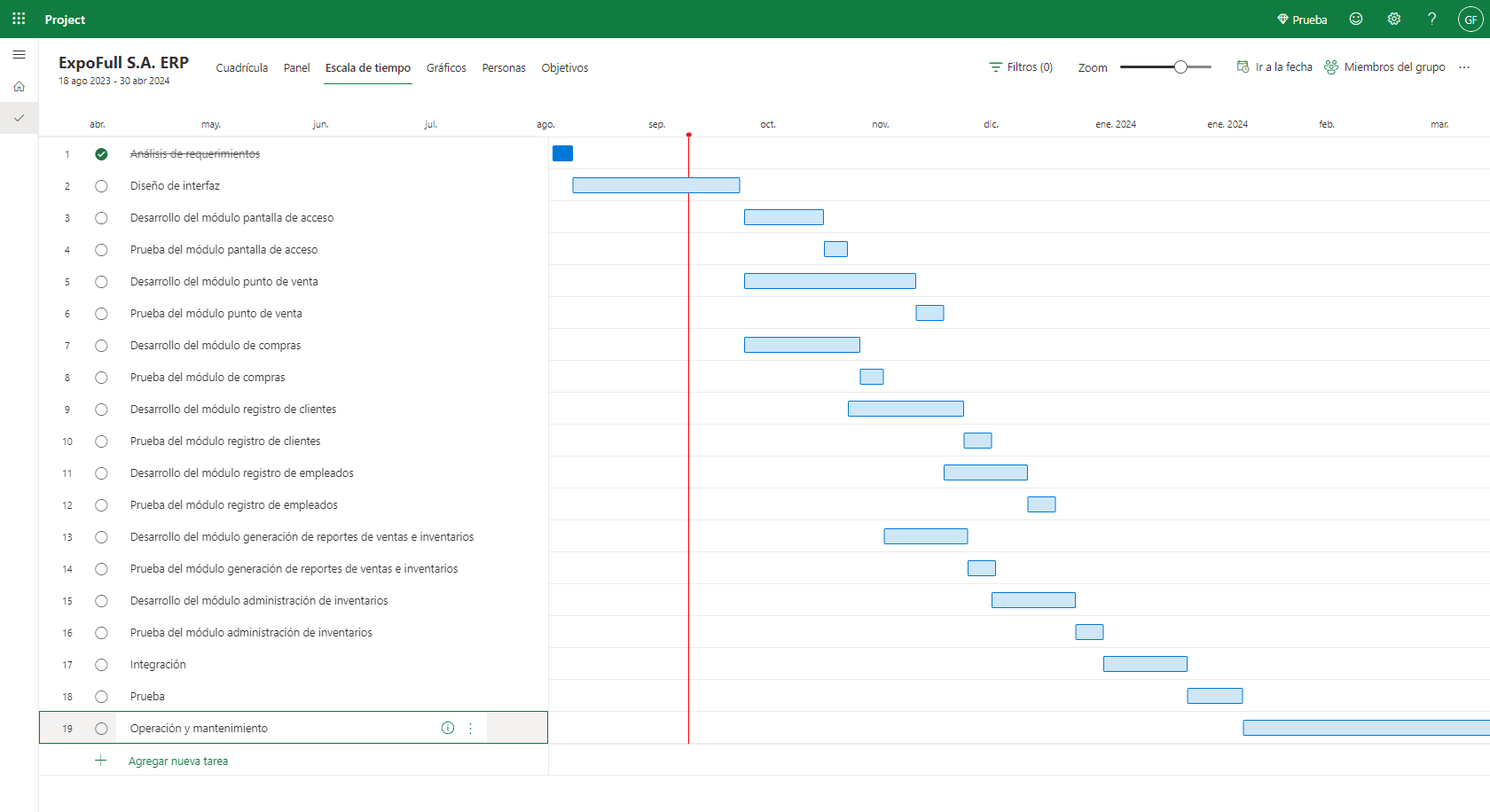
Administración de inventarios:

* Requerimientos de Hardware: Servidor, PC, Impresora, Escáner (De barras o de ondas RFID), Impresoras de etiquetas y Redes.
* Requerimientos de Software: Sistema de gestión de inventarios, Software de escaneo y captura de datos, Integración con los otros sistemas.
* Informes necesarios: No muestra informes.
* Tiempo estimado de trabajo: 14 días
* Roles involucrados: Experto en Ventas, Analista, Diseñador(es) de interfases,

Desarrollador(es) de software, Tester, Validador.

Nota: En algunos casos se cuenta con más de 1 equipo para realizar tareas, por lo que es posible iniciar una nueva tarea mientras hay otra(s) en curso (Como se aprecia en la calendarización a continuación).

**Calendarización:**

 **Nota: Las tareas no están asignadas porque MS Project permite asignar si existe un usuario registrado, no con usuarios ficticios, pero se entiende el punto y no hay dudas.**

**Link de proyecto en MS Project:**

[Proyecto - ExpoFull S.A. ERP](https://project.microsoft.com/?org=coppeldefault.crm.dynamics.com#/taskgrid?projectId=027ff4f8-af76-43b3-8766-ab25a1b7a6a3)

**Control de versiones de software:**

Para llevar el control de versiones de software, se ha seleccionado Github, esto debido a su popularidad, lo que hace que sea conocido por muchos y permita una rápida integración por parte del entero grupo de los participantes. Otra razón es su fácil uso, basta con poco tiempo de navegación para identificar las principales funciones. Al ser tan popular, también permite encontrar apoyo de parte de la comunidad para resolver problemas que surjan durante el desarrollo. El link de la plataforma es [github.com](http://www.github.com).

**Conclusión**

En conclusión, la implementación de un plan de proyecto, la calendarización adecuada y la elección de un método de control de versiones de software son aspectos fundamentales para el éxito de cualquier proyecto de desarrollo de software.

Un plan de proyecto bien estructurado y detallado permite establecer objetivos claros, asignar recursos de manera efectiva y definir las actividades necesarias para alcanzar los resultados deseados. Además, la calendarización adecuada garantiza que se cumplan los plazos establecidos y se eviten retrasos innecesarios.

Por otro lado, el control de versiones de software es esencial para gestionar los cambios y mantener un registro completo de las modificaciones realizadas a lo largo del desarrollo del proyecto. Esto permite mantener un historial de versiones, facilitar la colaboración entre los miembros del equipo y revertir cambios en caso de ser necesario.

En conjunto, estos elementos contribuyen a mejorar la eficiencia y la calidad del proceso de desarrollo de software. Permiten una mejor organización, seguimiento y coordinación de las actividades, lo que ayuda a minimizar riesgos, optimizar recursos y cumplir con los requisitos establecidos.

Para la conclusión de esta actividad, me siento más preparado para atender situaciones que requieran de orden/planificación, incluso en asuntos cotidianos de índole personal.

**Referencias**

5 Softwares de control de versiones. (2020, 11 febrero). *Drauta SEIDOR*. <https://www.drauta.com/5-softwares-de-control-de-versiones>

i-smart.mx. (2023, 20 abril). *- i-Smart.Mx*. i-smart.mx - Terminales certificadas para todo tipo de negocio.

https://i-smart.mx/precio-terminal-punto-de-venta/?gclid=CjwKCAjwsKqoBhBPEiwALrrqiEyvqsp\_sWriwWDErnwqVqxEpsVnzHD6iFP1RYYwTXmGiXQLQar6AhoCxvgQAvD\_BwE

Pezua, S. L. (2023). ¿Qué es GitHub?. ¿Cuáles son sus ventajas y desventajas? *Steven Lizarzaburu Pezúa*.

https://stevenlizarzaburupezua.com/github-que-es-ventajas-y-desventajas/#Ventajas-de-Github