

Actividad | #2 | Plan de Trabajo Ingeniería de Software I

Ingeniería en Desarrollo de Software

TUTOR: Eduardo Israel Castillo García

ALUMNO: Alejandra Ibarra Carmona

FECHA: 27/09/2024

Índice

Introducción…………………………………………3

Descripción………………………………………….3

Justificación…………………………………………4

Desarrollo………………………………………….5

* Plan de proyecto…………………………5
* Calendarización …………………………7
* Elección de Software para

Control de Versiones…………………….12

Conclusión…………………………………………13

Referencias……………………………………….13

Introducción

En esta actividad ya comenzaremos a visualizar nuestro plan de trabajo para el desarrollo de nuestro software del cliente ExpoFull SA. Una vez que se haya elegido el método de desarrollo de software se deben clasificar ciertos temas:

* Alcance del proyecto
* Plan de trabajo
* Hitos del proyecto
* Calendarización
* Costos del proyecto

Nuestro alcance deberá estar bien limitado, en este paso se deben indicar datos concretos sobre nuestro proyecto y aquellas limitaciones que nos pudiera surgir y facilidades en las que se va a disponer, hay cinco aspectos que se deberían cumplir en esta etapa y son los siguientes:

* Función: Se revisará y se concentraran las descritas en la ingeniería
* Rendimiento: Se basa en las necesidades en cuanto al tiempo de respuesta y el procesamiento
* Restricciones: Son las limitaciones de hardware con las que se encontrara nuestro software a desarrollar
* Interfaces: se deben identificar la información que se comunicara a los dispositivos, así como los elementos del software y el personal humano a ocupar
* Fiabilidad: Se establece el nivel de seguridad que debe tener el proyecto dependiendo su naturaleza

Descripción

Después de ya realizado nuestro alcance de proyecto podemos continuar con nuestro plan de proyecto, que es algo que nos pide esta segunda actividad veamos como es que tiene que estar compuesto:

* Introducción: Describe los objetivos del proyecto y expone restricciones que afecten la gestión de nuestro proyecto.
* Organización del proyecto: Describe la forma en la que los integrantes están organizados y como se encuentran involucrados.
* Análisis de riesgo: Describe la probabilidad de que existan riesgos y que se encuentren estrategias para reducirlos.
* Requerimientos de hardware y software: nos describe el hardware y el software requeridos para llevar a cabo el desarrollo
* División de trabajo: Describe la división del proyecto en actividades e identifica los hitos y productos a entregar asociados con el producto
* Programa del proyecto: Describe las dependencias de las actividades y el tiempo estimado requerido para alcanzar cada hito y asignación de personas
* Mecanismos de supervisión e informes: En este paso es la gestión de informes y fechas, así como los mecanismos de supervisión.

Justificación

Ahora como en varios puntos anteriores mencionamos que se tienen que cumplir con los hitos y estos se refieren a los momentos clave en los que se deben realizar las entregas concretas, estos a veces representan el fin de una etapa lógica en el proyecto. Para establecer los hitos en el proceso de software deben dividirse en actividades básicas con sus salidas asociadas, una vez que ya se tienen definidos se debe establecer la calendarización del proyecto esta es una de las tareas más difíciles para los gestores de proyectos ya que ellos estiman el tiempo y los recursos que se requieren para completar las actividades y organizarlas de forma coherente. A veces dar una estimación aún se complica un poco mas ya que los diferentes proyectos pueden utilizar métodos de diseño y lenguajes de programación diferentes.

Esto nos implica separar todo el trabajo de un proyecto en sus actividades complementarias y así poder considera el tiempo para completar dichas actividades:

* Identificar actividades (requerimientos del software)
* Identificar dependencias de actividades
* Estimar recursos para las actividades
* Asignar personas a las actividades
* Crear gráficos de proyectos (redes de actividades y graficas de barras) de aquí se puede regresar al paso tres.

Desarrollo

* Plan de trabajo

|  |  |
| --- | --- |
| Visión General | La empresa IT Power desarrollara un software para la empresa ExpoFull. SA el cual le ayudara a tener mas segura la informacion de sus clientes ya que contara con 50 puntos de venta a nivel nacional. El ERP nos permite agilizar y automatizar procesos y gestión de recursos ya que nos proporciona una vista completa de todas las operaciones comerciales. |
| Alcances | * Pantalla de Acceso * Punto de Venta * Módulo de Compras * Registro de Clientes * Registro de Empleados * Generación de reportes de compras, ventas e inventarios. * Gestión de inventarios |
|  |  |
| Objetivos | * Desarrollar un sistema el cual facilite el uso a ExpoFull y agilice los procesos de actualización de inventario. * Mantener la conexión en una red segura y exclusiva para evitar perdida de información o que falle el sistema. * Crear un catalogo virtual para dentro del software para verificar modelos disponibles * Seguimiento PostVenta para cambios y devoluciones |

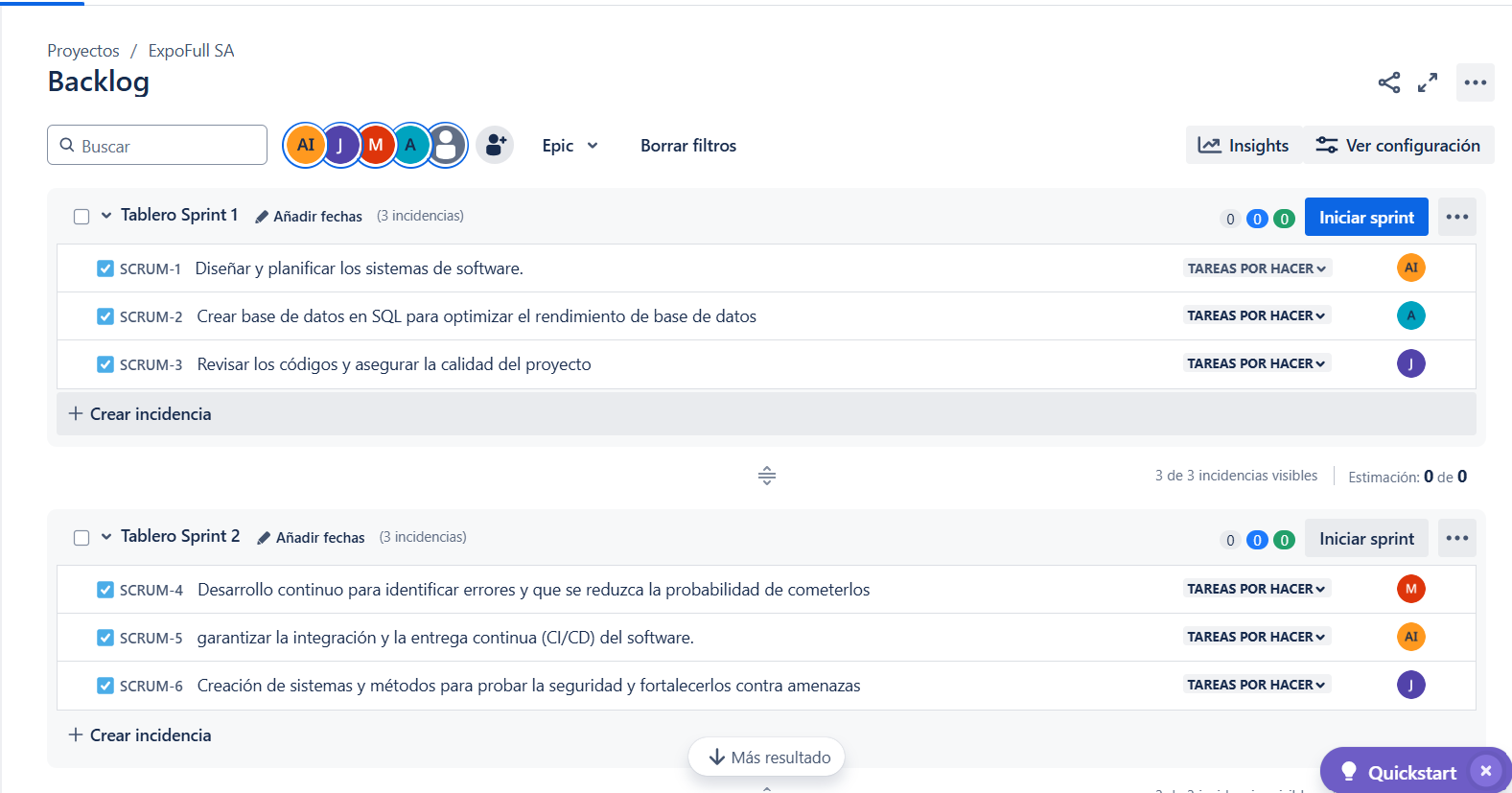
* Requerimientos de Software y Hardware para el desarrollo

|  |  |
| --- | --- |
| Hardware | \*Puntos de venta móviles, es decir tablet con impresoras para una venta ágil  \*Computadoras con una RAM de 512 GB para un funcionamiento correcto.  \*Procesador Pentium 4 de 1 GHz con 256kb de cache.  \*1 placa Fast Ethernet 10/100 |
| Software | \*Sistema operativo diseñado para funcionar en Windows  \*Para la base de datos utilizar MYSQL ya que nos ayuda a centralizar la información, estructurar los datos en la tabla, campos definidos e indices.  \*Frontend se utilizará HTML (lenguaje de marcado de hipertexto) en Visual Studio Code  \*Para el Backend en donde se desarrollará la logica para que nuestro sitio funcione y el lenguaje que ayudara a su desarrollo será Python\*  Desarrollo de base de datos |

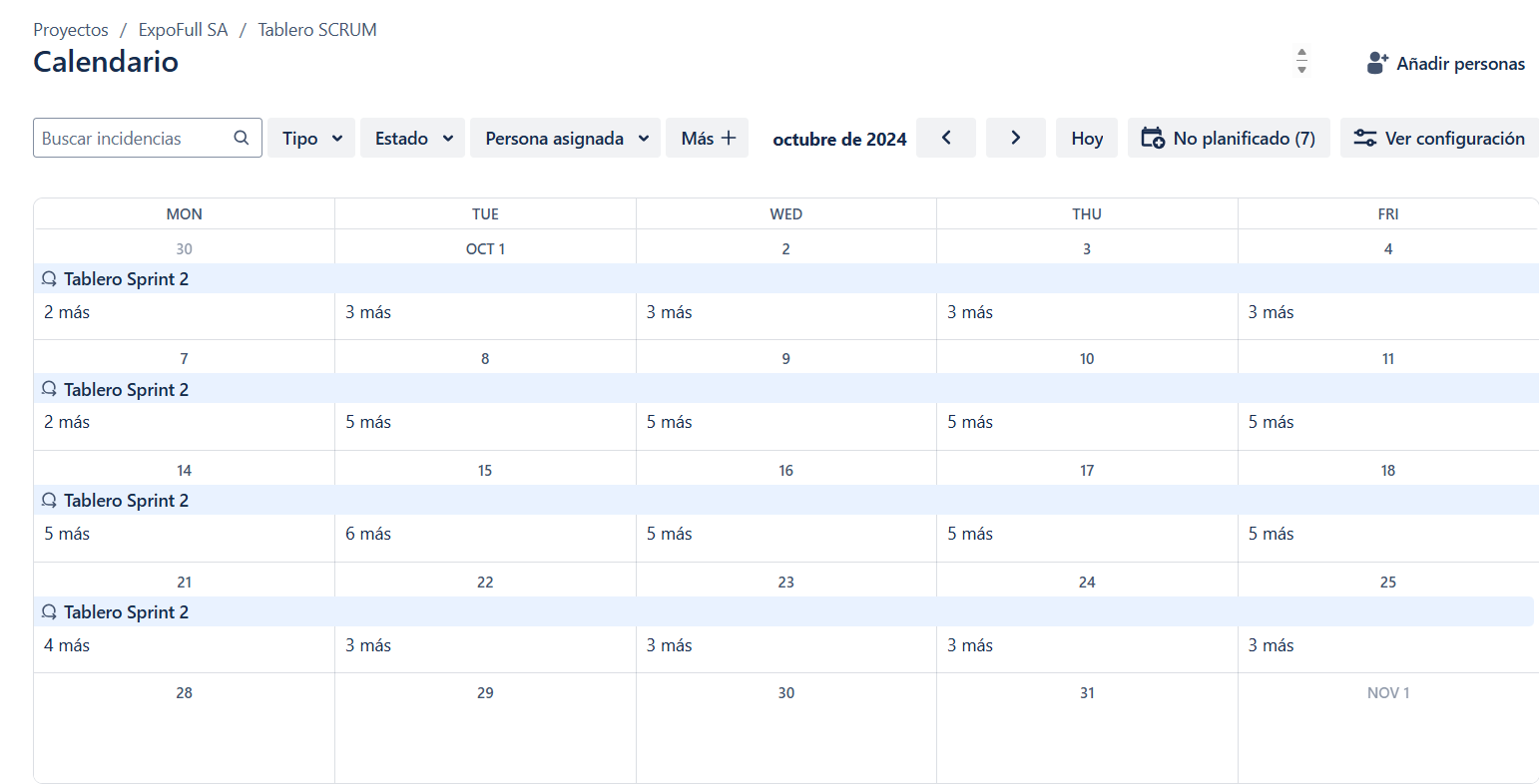
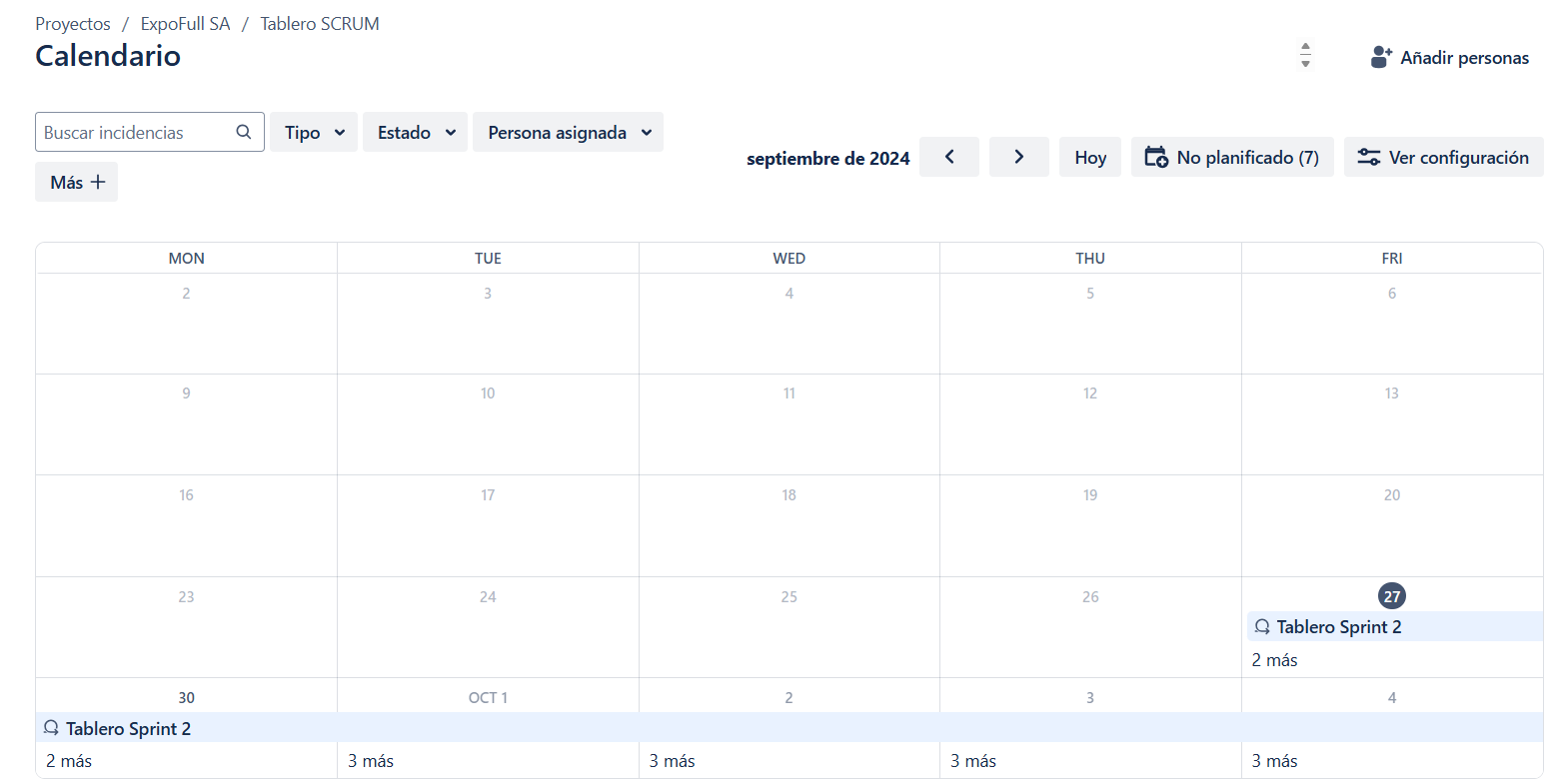
* Roles que se designaron para el desarrollo

|  |  |  |
| --- | --- | --- |
| Miembro del equipo | Roles | Responsabilidades |
| Alejandra Ibarra | Líder del proyecto | * Representante del proyecto que define las prioridades de las tareas |
| Miguel Miranda | Desarrollador base datos | * Crear base de datos en SQL para optimizar el rendimiento de base de datos |
| Josué García | Arquitecto de desarrollo | * Revisar los códigos y asegurar la calidad del proyecto * Diseñar y planificar los sistemas de software. |
| Irene Ibarra | Ingeniero DevOps | * Trabajan en mejoras del sistema y estrategias de despliegue y que garantizan la integración y la entrega continua (CI/CD) del software. * Desarrollo continuo para identificar errores y que se reduzca la probabilidad de cometerlos |
| Juan Jiménez | Desarrollador de seguridad | * Creación de sistemas y métodos para probar la seguridad y fortalecerlos contra amenazas |
| Adrián Estrada | Tester | * Pruebas del funcionamiento del sistema |

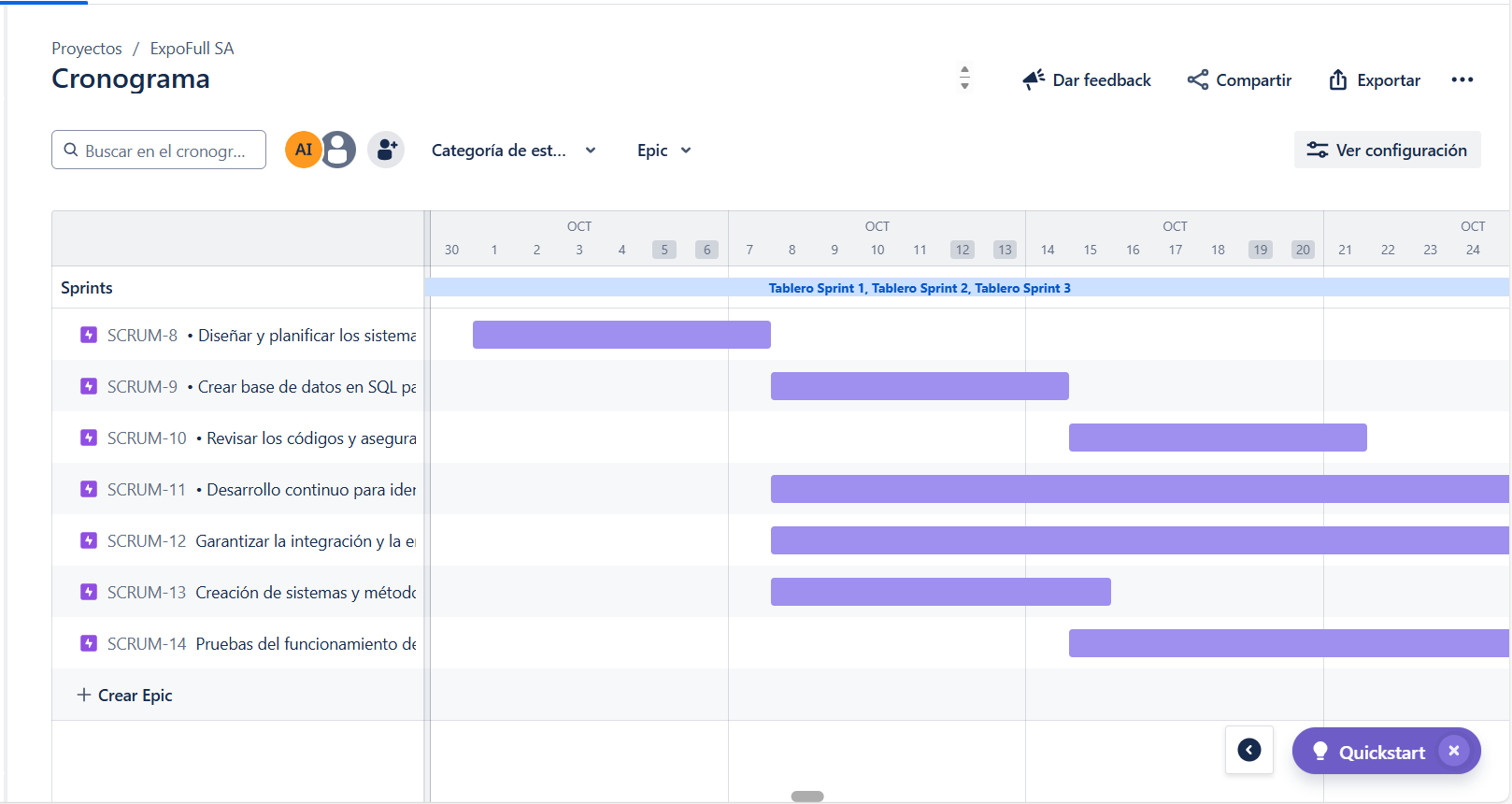
* Calendarización del proyecto



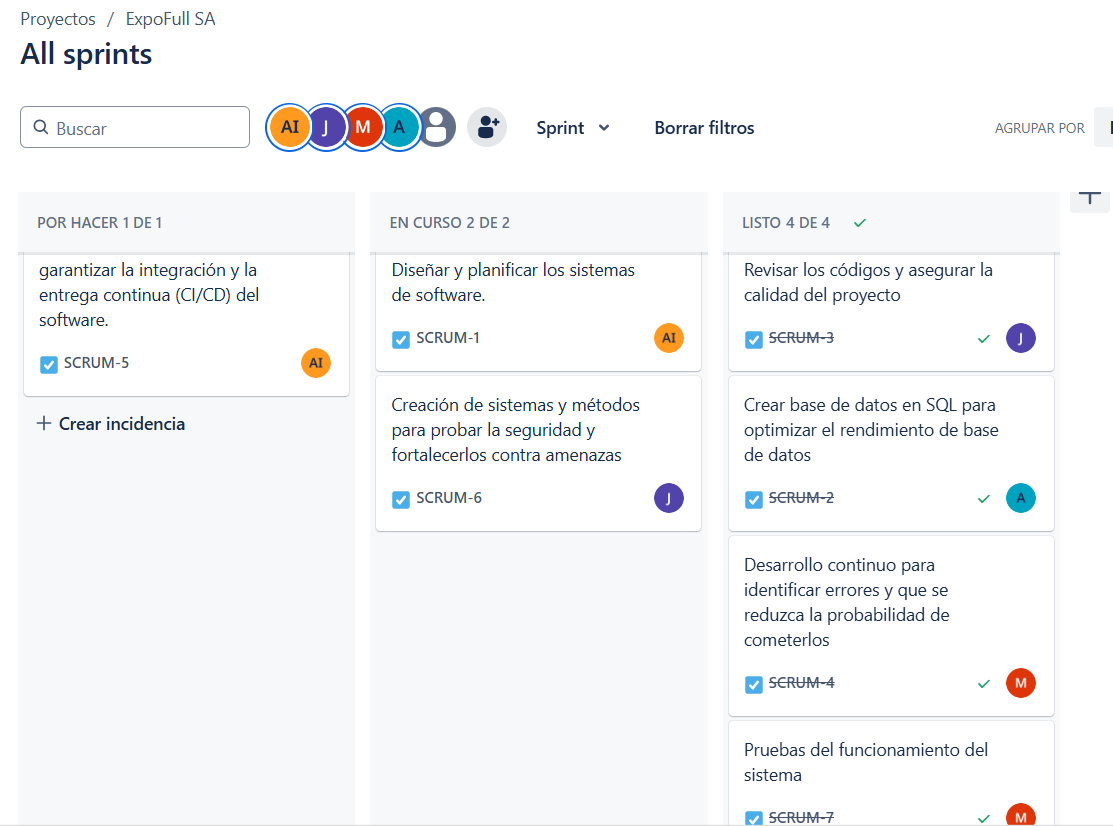
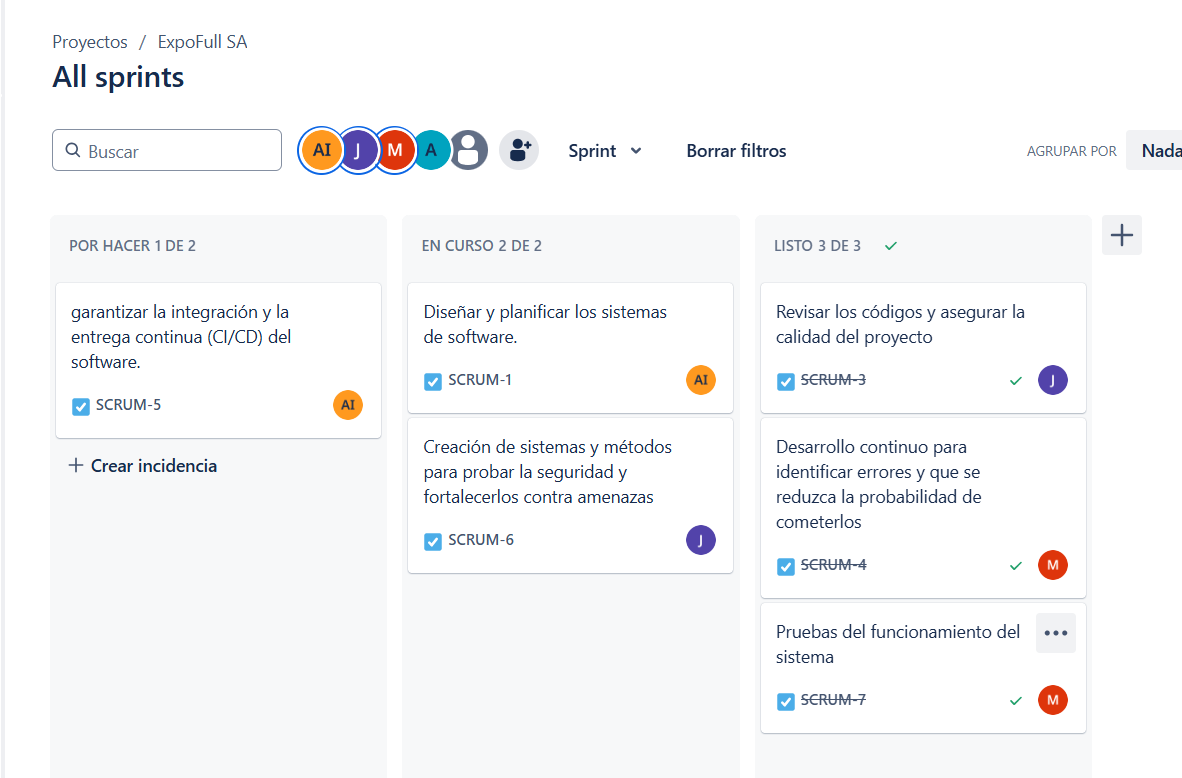
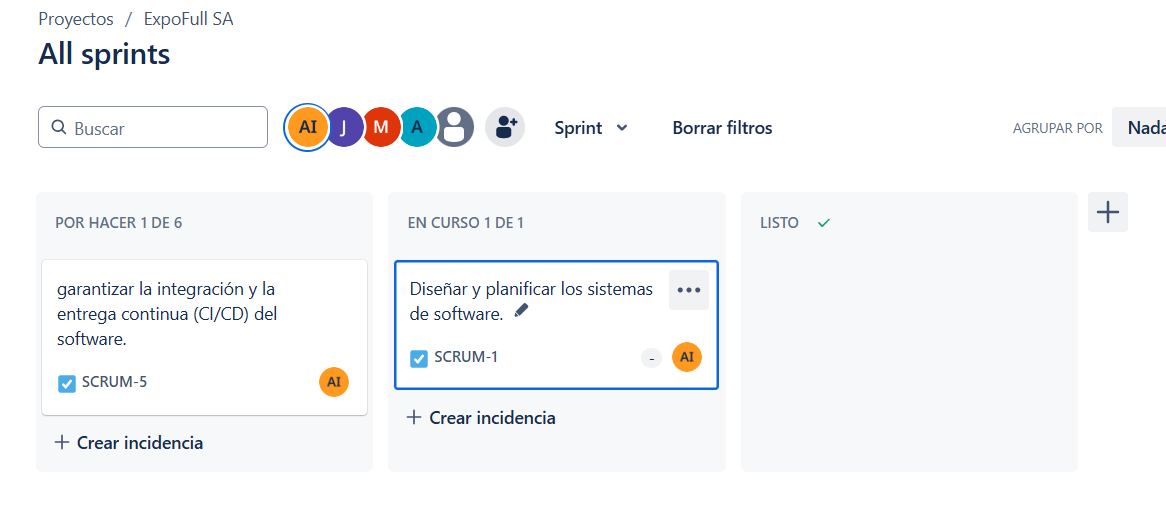
Se puede observar que se crean las actividades en el backlog y se asignann los responsables para la que se ejecuten y se terminen en tiempo y forma, una vez realizados se crean los sprit para la hacer una entrega parcial del proyecto al cliente



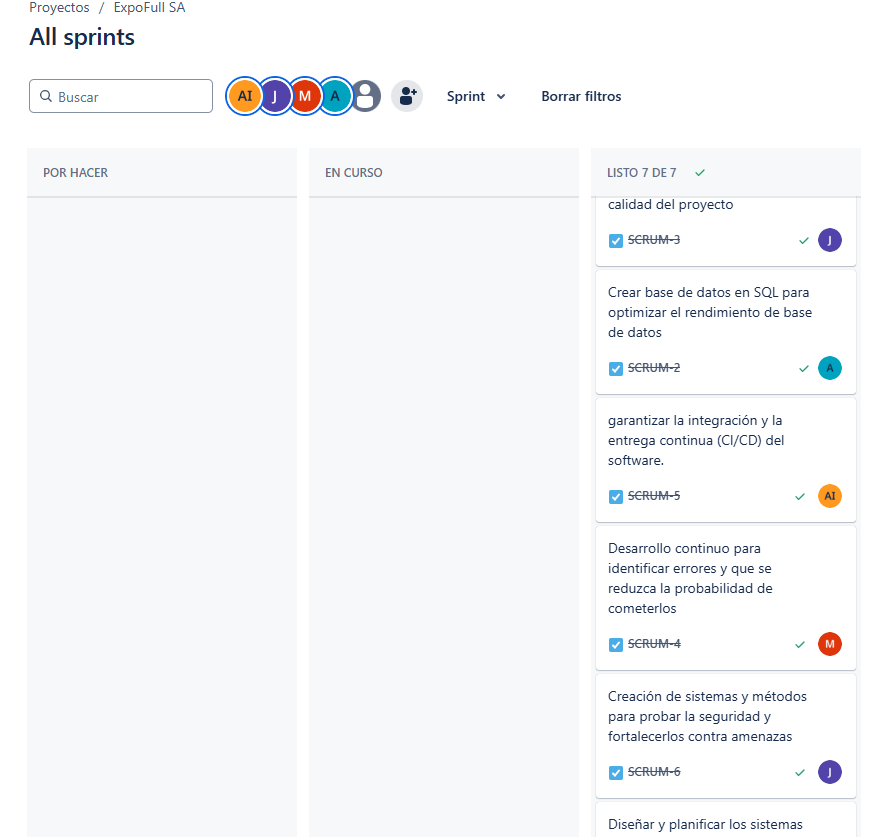
* Cronograma de actividades



* Sprint



Se finalizan todas las actividades del proyecto



* Elección del software para el control de versiones

El control de versiones también conocido como control de código fuente, es la practica de rastrear y gestionar los cambios en un software a lo largo del tiempo. A medida que los entornos de desarrollo se aceleran el control de versiones nos ayudan a trabajar de forma más inteligente y rápida.

Algunas de las ventajas:

* Cambios a largo plazo en todos los archivos, esto incluye todos los cambios que cualquier persona realice. Tener un historial completo permite volver a las a las versiones anteriores para ayudar a analizar la raíz de los errores.
* Creación de ramas y fusiones: la creación de las ramas em las herramientas VSC nos ayuda a mantener el flujo del trabajo independientes unos de los otros, es decir que son independientes a lo que otra persona pueda hacer.
* Trazabilidad: trazar cada cambio que se hace en el software conectarlo con uno de gestión de proyectos y seguimiento de errores como jira y además anota cada cambio con un mensaje que describa el propósito y el objetivo del cambio.

El software que utilizare en este proyecto será GitHub ya que ofrece muchas ventajas que nos ayudan a que alguien mas pueda compartir sus puntos de vista sobre nuestro trabajo que subimos a la plataforma, también se pueden integrar algunos otro y cada cambio se registra lo que nos permite volver a versiones anteriores de nuestros trabajos. Esta funcionalidad que tiene es clave cuando se trabaja en proyectos colaborativos como este, donde se definen roles y entrega de actividades.

Conclusiones

Podemos concluir que el método de trabajo scrum es una forma más rápida y sencilla de entregar proyectos, también las metodologías agiles nos ayudan a reducir la probabilidad de fracaso por medio de dos aspectos fundamentales que son retrasar las decisiones y la planificación optativa, sus fundamentos se basan en la adaptabilidad de los procesos de desarrollo.

Sus características principales son:

* Comunicación
* Cohesión
* Funcionalidad
* Conocimientos

El proceso de scrum se aplica de manera regular y es un conjunto de buenas prácticas para trabajar colaborativamente y que se tenga un mejor resultado en equipo, se realizan entregas parciales y regulares del producto final que se priorizan por el beneficio que aporta cada receptor del equipo. Se utiliza para resolver situaciones en las cuales no se a entregado al cliente lo que necesita, y una cosa importante cuando las entregas se alargan demasiado los costes se disparan y la calidad no es aceptable.