

Específicas

- **Competencia: Analiza la evolución disciplinar de la Ingeniería de Software, así como las características del profesionalista vinculado con la misma, de acuerdo con el marco teórico y los modelos curriculares de la disciplina.**

Una vez visto en clase la evolución de las metodologías, las tradicionales y las ágiles, se implementa una metodología ágil llamada scrum que es resultado de muchos años de cambio dentro del mundo del software, se realizaron sprints de trabajo, en los cuales se definieron objetivos específicos, se revisaron las tareas completadas y se ajustaron los planes según las revisiones. Esta práctica no solo refleja las tendencias actuales en la Ingeniería de Software, sino que también muestra cómo el ingeniero moderno debe combinar habilidades técnicas con capacidades de gestión de proyectos para coordinar eficientemente los esfuerzos del equipo. También mediante el trabajo colaborativo se pudo reflejar que la interacción constante entre diferentes roles es esencial para el éxito del proyecto, lo cual es un punto muy importante en la ingeniería de software moderna.

- **Competencia: Analiza los principales métodos, técnicas, procedimientos y buenas prácticas utilizados en las fases de requisitos, diseño, codificación, pruebas y mantenimiento del software, de acuerdo con el cuerpo de conocimientos reconocido por la disciplina.**

Podemos identificar fases clave como diseño y la fase de requisitos, durante esta segunda entrega consideramos que nos encontramos más en la fase de diseño, aunque de igual forma concluimos parte de la fase de requisitos. En esta fase logramos reorganizarla y mejorarla, cambiando algunos y seleccionando a los más importantes, esto mediante el método de MosCow, además utilizamos una técnica para plasmar los requisitos (historias de usuario) para poder usarlos dentro de nuestro método ágil (Scrum)

Definiendo procedimiento como los conjuntos de pasos o acciones sistemáticas que se siguen para llevar a cabo una tarea específica, resolver un problema o lograr un objetivo. Podemos decir que algunos de nuestros procedimientos para esta segunda entrega son:

Planificación de Sprints: Definición de ciclos de trabajo cortos y metas específicas dentro de nuestro método ágil

Creación de las historias de usuario: Usamos una serie de pasos, refinar los requerimientos, adjuntarles la estructura indicada de las historias, agregarles priorización y agregarles los criterios de aceptación

Creación del backlog: Investigar diseños con los que estemos cómodos trabajando, plasmarlo a github y plasmar las historias de usuario dentro

Creación del bosquejo: La del bosquejo conllevo una serie de pasos a seguir, asignar un equipo, asignarle tareas al equipo y validar su trabajo

- **Competencia: Selecciona el Modelo de Ciclo de Vida del Software en el inicio de un proyecto, con base en las características del equipo de desarrollo, del cliente y de la problemática.**

Seleccionar un modelo de ciclo de vida adecuado en la fase inicial del proyecto es esencial para gestionar eficazmente el desarrollo, considerando las características del equipo, el cliente y la problemática. Para nuestro proyecto, se eligió SCRUM, un marco de trabajo ágil, que ayudan al equipo a empezar a adaptarse a las nuevas tendencias de la ingeniería en software

SCRUM en nuestro proyecto:

La estructura de SCRUM permite definir claramente roles como el Product Owner, Scrum Master, y el equipo de desarrollo, en nuestro proyecto esta competencia se plasmó de forma que mediante las características de nuestros miembros decidimos asignarles estos puestos, aunque realmente no lo hicimos de la mejor forma, debido a imprevistos Cauch Salazar Samuel tomó posición de scrum master y se le asignó la tarea de dirigir el backlog (Tarea que le pertenecía al producto owner). De igual forma intentamos seguir el ciclo de vida lo más correcto posible, estableciendo sprints, dirigiendo el backlog, haciendo reuniones para compartir avances y problemáticas.

- **Competencia: Áreas de Gestión Software. Analiza los principales métodos, técnicas, procedimientos, y buenas prácticas utilizados en los procesos de estimación, planificación, seguimiento, control, calidad y configuración del software, de acuerdo con el cuerpo de conocimientos reconocido por la disciplina**

La competencia se refleja de la siguiente forma:

Método de Planificación Ágil (SCRUM): Utilizar SCRUM permitió al equipo realizar estimaciones en cada sprint, priorizando las funcionalidades más valiosas para el desarrollo del proyecto

Estimación de Tareas: El equipo pudo emplear técnicas de estimación sobre las Historias para determinar la complejidad de cada tarea, favoreciendo una planificación más precisa.

Un control bueno sobre el progreso y la calidad de cada funcionalidad desarrollada, mediante pequeñas reuniones en la escuela y una comunicación digital

Competencia: Factores Humanos en Ingeniería de Software. Identifica los principales factores humanos que inciden en el éxito o fracaso de los procesos de software, de acuerdo con el marco teórico de la disciplina.

Una competencia muy presente dentro de nuestra entrega, debido a que muchos de los miembros tuvieron actividades escolares y extraescolares que perjudicaron a un desarrollo constante del proyecto, de igual forma por problemas familiar y de salud el proyecto fue tomando bajas que retrasaron grandemente la entrega, de forma que un buen liderazgo fue clave para sacar a flote el proyecto. Un factor humano muy importante es la motivación, presentes en algunos miembros, fue clave para el éxito de la entrega.

El éxito del proyecto del mapa interactivo de la facultad depende en gran medida de factores humanos, como la comunicación, colaboración, adaptabilidad, habilidades y liderazgo. Como nota adicional como equipo consideramos que las historias de usuario son de gran utilidad para la motivación del equipo, aportan contexto al equipo, ayudando a que comprendan la importancia y el impacto de su trabajo. Esta comprensión contribuye a una mayor motivación, ya que el equipo ve cómo sus tareas específicas se relacionan con el objetivo general del proyecto y con la satisfacción del usuario final.

Genéricas

· **Competencia: Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma**

Se promueven en distintos momentos durante el desarrollo del producto, al organizar el equipo, al dar instrucciones sobre actividades específicas, la capacidad de comunicar es crucial para una buena coordinación del equipo, expresar ideas, soluciones e incluso problemas hace que el proyecto tome un buen ritmo y evita estancamientos dentro del mismo. Muchas veces se presentaron situaciones en donde miembros del equipo no tenían clara su tarea por una mala comunicación, eso nos hizo aprender que los mínimos detalles en la comunicación de forma oral o escrita hace una diferencia brutal en el mensaje.

Comunicación en forma oral:

Tuvo gran presencia, especialmente dentro de nuestras reuniones y en el Sprint Review junto con el profesor, ya que es el momento indicado donde necesitamos comunicarnos de la mejor manera para explicar nuestro avance y principalmente para expresar nuestras dudas.

Comunicación en forma escrita:

Podemos resaltar que se incita a redactar de forma coherente y correcta, esto presentado en la documentación y en las historias de usuario, debido a que el equipo las leerá y podrá ser de gran utilidad para que personas ajenas al proyecto tengan un contexto de este.

- **Competencia: Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable**

Se utilizaron herramientas de colaboración y software de gestión de proyectos, como Lucidchart y Git Hub, utilizándolas de forma pertinente logramos crear el bosquejo de algunas interfaces. Muchos de los miembros utilizaron distintas para documentar (Word, Notepad, ...) y presentarlo en un lenguaje de marcado (Marckdown)

- **Competencia: Gestiona el conocimiento en sus intervenciones profesionales y en su vida personal, de manera pertinente.**

Se organizaron reuniones para compartir conocimientos sobre herramientas como git o lucidchart, en donde alguno de los miembros del equipo ayudo a los demás, gracias a esa gestión del conocimiento, la gestión fue pertinente porque mejoró la calidad del producto, facilitó la comunicación, el trabajo y dejó un aprendizaje valioso para el equipo, tanto a nivel profesional como personal.

Filtro. La posibilidad de filtrar información en el mapa también puede ser vista como una forma de gestionar el conocimiento del usuario, permitiendo una experiencia más personalizada. Además, para gestionar el conocimiento de forma eficaz surgieron artefactos que ayudarán a los miembros y posibles miembros futuros a contextualizarse sobre el proyecto, se plasmó dentro del repositorio, de modo que cualquier miembro puede consultar la información durante el desarrollo.

- **Competencia: Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa**

Se fomenta un ambiente de trabajo cooperativo entre los miembros del equipo, durante la fase de diseño del proyecto, el equipo se conformó por diferentes miembros con habilidades especiales, algunos resaltando en gestión, otros en redacción y otros en el uso de herramientas para el bosquejo, todos ellos haciendo un trabajo colaborativo con el objetivo de cumplir con la segunda entrega, pero principalmente completar el proyecto.

Esta competencia resalta más en la gestión del backlog, donde se asignaron responsabilidades claras y se registró el estado de cada tarea, facilitando la coordinación y permitiendo que cualquier miembro pudiera apoyar a otro si una tarea se retrasaba.

· **Competencia: Toma decisiones en su práctica profesional y personal, de manera responsable**

Durante el desarrollo del proyecto, el equipo se encontró con una decisión importante: integrar funcionalidades avanzadas que permitirían a los usuarios recibir notificaciones sobre cambios en sus horarios de clase. Sin embargo, estas funcionalidades requerían tiempo adicional y nos replanteamos si realmente era algo que queríamos agregar a nuestro proyecto, decidimos eliminar esta funcionalidad considerando que no afecte a otras y corroborando que no sea esta funcionalidad no sea una prioridad para el objetivo. Se presentaron otras situaciones así, donde el resultado fue la eliminación de diferentes historias de usuarios y requerimientos, siempre priorizando que el equipo actué de manera responsable considerando diferentes variables como el tiempo, la capacidad y el trabajo.