

# **LA METODOLOGÍA SCRUM COMO ESTRATEGIA DE DESARROLLO DE SOFTWARE EDUCATIVO.**

## **I. Introducción**

El desarrollo de software educativo enfrenta desafíos únicos que requieren un enfoque flexible y colaborativo. En este contexto, las metodologías ágiles han ganado relevancia, permitiendo a los equipos adaptarse rápidamente a las necesidades cambiantes de los usuarios y a los requisitos del proyecto. La metodología SCRUM se ha consolidado como una herramienta eficaz para gestionar proyectos complejos, facilitando la organización del trabajo y promoviendo una comunicación continua entre los miembros del equipo.

SCRUM se basa en principios fundamentales que enfatizan la colaboración, la transparencia y la adaptación. Según Schwaber y Sutherland (2017), "SCRUM es un marco que permite a las personas abordar problemas complejos y entregar productos de valor más alto". Este enfoque es especialmente relevante en el ámbito educativo, donde la interacción constante entre docentes y estudiantes es crucial para el éxito del software desarrollado.

El proceso de desarrollo se estructura en ciclos cortos llamados "sprints", lo que permite a los equipos realizar entregas frecuentes y recibir retroalimentación continua. Esta dinámica no solo mejora la calidad del producto final, sino que también fomenta un ambiente de aprendizaje donde los integrantes pueden compartir conocimientos y

resolver problemas de manera conjunta. Como menciona Sutherland (2014), "la colaboración en equipo es esencial para el éxito en SCRUM".

El desarrollo de software educativo implica una serie de etapas críticas que abarcan desde la propuesta inicial hasta la implementación final. Cada fase del proyecto se beneficia de un enfoque ágil, permitiendo ajustes rápidos basados en la retroalimentación obtenida durante las revisiones periódicas.

El uso de herramientas modernas como GitHub para la gestión del código fuente y Google Docs para la colaboración en documentos ha permitido a los equipos trabajar de manera más eficiente. Estas plataformas no solo facilitan la organización del trabajo, sino que también promueven una cultura de transparencia y responsabilidad compartida entre los miembros del equipo. El desarrollo de software educativo no solo optimiza el proceso de creación, sino que también potencia las habilidades técnicas y colaborativas del equipo. Este enfoque integral asegura que el producto final no solo cumpla con los requisitos técnicos, sino que también responda efectivamente a las necesidades de los usuarios finales.

## **II. Metodología**

El desarrollo del software educativo se estructuró en tres etapas correspondientes a los cursos de diseño de software educativo I, II Y III. En cada etapa, se aplicaron metodologías ágiles para organizar las tareas, fomentar la colaboración y garantizar la calidad del producto final. A continuación, se describen las actividades realizadas en cada curso, los roles asumidos, las tecnologías utilizadas y los desafíos enfrentados.

### **Diseño de software educativo I: propuesta y diseño técnico del software.**

En el primer curso de software, el objetivo principal fue definir la propuesta del diseño del software educativo. El equipo de trabajo propuso un componente de contenido que le permitiría a los docentes subir información y a los estudiantes visualizar, descargar, compartir y comentar dicho contenido. Este proceso incluyó la creación de un documento con todos estos requerimientos:

Definiciones y acrónimos: se estableció un glosario común para asegurar un entendimiento uniforme de los términos utilizados en el proyecto.

Descripción general: se identificaron a los usuarios del sistema y sus necesidades.

Usuarios: docentes como gestores de contenido y estudiantes como consumidores de información.

Restricciones: incluyeron limitaciones técnicas y operativas que debían ser consideradas en el diseño.

Requisitos funcionales: se elaboraron casos de uso con descripciones detalladas,

diagramas de flujo y priorización de requisitos para estructurar las funcionalidades esenciales del sistema.

Requisitos no funcionales: documentamos especificaciones relacionadas con desempeño, seguridad, usabilidad y escalabilidad, fundamentales para la calidad del software.

Este curso se enfocó en todo el proceso de documentación técnica necesaria para poder desarrollar ese software, sirviendo como base para las etapas posteriores. Para el desarrollo de la propuesta desde un inicio y por asesoría del docente tutor se optó por la metodología SCRUM debido a la flexibilidad y enfoque que esta brinda. Para el trabajo en equipo se realizaron reuniones semanales y revisiones continuas, en las cuales se dividió el desarrollo de sprints, donde se discutían los avances, desafíos y búsqueda de soluciones a ellos. Estas reuniones fueron llevadas a cabo por medio de Meet. Esto con el fin de mantener un enfoque interactivo, también se creó una bitácora donde se llevó de manera detallada todas las actividades y modificaciones que se realizaban, cabe resaltar que estas eran supervisadas y revisadas por el docente, además de que cada miembro del equipo asumió roles específicos.

### **Diseño de software educativo II : desarrollo de la infraestructura y base de datos.**

En el segundo curso de diseño de software, el enfoque principal fue adquirir y aplicar conocimientos teóricos y prácticos relacionados con el desarrollo del backend, esenciales para la implementación de la infraestructura del sistema. Durante esta etapa, se abordaron aspectos fundamentales como la configuración de servidores, la

integración de servicios y el manejo eficiente de datos, estableciendo las bases técnicas necesarias para el funcionamiento del software educativo. El equipo trabajó con herramientas modernas que facilitaron el desarrollo de la infraestructura. MongoDB se utilizó para gestionar la información del sistema, diseñando esquemas y relaciones que soportaran las funcionalidades del software. Por otro lado, NestJS permitió estructurar la lógica del servidor y las interacciones entre el frontend y el backend, garantizando un flujo eficiente de datos.

El proceso incluyó varias actividades clave. En primer lugar, se diseñaron bases de datos que soportaran las operaciones necesarias, como enviar, recibir, actualizar y eliminar contenido, respondiendo a las necesidades de los dos tipos de usuarios definidos previamente: docentes y estudiantes. También se configuraron servidores para establecer la comunicación entre los distintos componentes del sistema. Finalmente, se trabajó en la integración de servicios, probando y ajustando las funcionalidades para asegurar su correcto funcionamiento.

Durante el desarrollo, se presentaron diversos desafíos técnicos, principalmente relacionados con la integración de tecnologías y la implementación de las primeras conexiones entre componentes. Estos retos se superaron mediante la consulta a expertos externos, la retroalimentación del docente encargado del curso y el aprendizaje autodidacta del equipo. Estas estrategias permitieron identificar soluciones efectivas y mejorar continuamente.

Al término del curso, el equipo logró integrar de manera eficiente la estructura y el backend del software. Esta experiencia no solo fortaleció las habilidades técnicas de los integrantes, sino que también permitió aplicar los conocimientos adquiridos en un contexto práctico, sentando una base sólida para las siguientes fases del proyecto.

### **Diseño de software educativo III: implementación y diseño de interfaz.**

En el curso final de software, se materializó la propuesta del componente de contenido mediante el desarrollo de interfaces funcionales y el inicio de la programación del sistema. Los principales logros fueron:

La elaboración del diagrama de contenido, el guión técnico, el guión didáctico y la guía de metáforas, lo que permitió conceptualizar y estructurar la navegación y la experiencia de usuario. Se diseñó un mapa de navegación detallado para garantizar la fluidez en el uso del software. Se implementaron las primeras ventanas del sistema, incorporando funcionalidades clave como la visualización y el manejo del contenido. El trabajo se organizó utilizando SCRUM, con roles específicos: Diseño de interfaz: desarrollo de prototipos y front-end. Ingeniería de pruebas: verificación y validación del sistema en cada iteración. Programadores: implementación del backend y la lógica del software. La colaboración se centralizó en GitHub, donde se gestionó el código fuente, las revisiones y los aportes individuales. Aunque hubo retos técnicos, como problemas de integración y diseño, la metodología ágil permitió resolverlos de manera eficiente, por otro lado, el docente encargado brindó un espacio de aula invertida, donde como grupo se debía ir avanzando de forma eficaz en nuestro proyecto.

Se fijaron los objetivos y entregas semanales de nuestro proyecto. esto con la finalidad de optimizar el desarrollo de nuestro trabajo y poder obtener retroalimentaciones del docente para realizar un proceso de ajuste y mejora en nuestro software. Gracias a la metodología Scrum logramos realizar discusiones retroalimentadas en grupo las cuales nos brindaron una oportunidad para mejorar la comunicación, fomentar la colaboración y encontrar soluciones.

La experiencia adquirida en estos tres cursos no sólo consolidó las habilidades técnicas, sino que también mejoró la capacidad de trabajo en equipo, organización y resolución de problemas. En este curso, se aplicaron todos los conocimientos adquiridos para realizar el diseño e implementación del frontend, desarrollando interfaces funcionales y programando las primeras ventanas del sistema. Esto incluyó acciones claves como la visualización y manejo del contenido, basados en el diagrama, un guión técnico y el mapa de navegación que se estructuró previamente en el curso de diseño de software educativo I.

Se realizaron encuentros semanalmente de manera asíncrona dentro de la plataforma Meet con la participación de todos los integrantes del grupo, por lo general estos encuentros duraban alrededor de dos horas, en este espacio, se mostraban y explicaban los avances realizados por cada integrante, se realizaban discusiones, retroalimentación, entre otros aspectos. El uso de la plataforma Google Doc nos permitió subir documentos y trabajar de manera colaborativa en la construcción y

desarrollo de la documentación y diseño de actividades en pro al desarrollo de nuestro proyecto.

Se utilizaron herramientas de diseño y programación, se materializó la navegación del sistema y las funcionalidades del software, garantizando una experiencia de usuario fluida. Gracias a la metodología SCRUM se organizaron los roles para el diseño de interfaces, pruebas de testeo las cuales permitieron verificar el funcionamiento y navegación del software, y por último los roles de programación. Esta metodología permitió trabajar de manera colaborativa en la construcción del proyecto por medio de la plataforma GitHub.

Este trabajo fue un proceso colaborativo entorno a nuestro grupo de trabajo, realizamos una distribución en las áreas de trabajo en la que mejor se desenvolvía cada integrante, quedando de la siguiente manera: dos integrantes del grupo quedaron como programadores, otros dos como diseñadores de interfaz y uno como encargado de pruebas de testeo del software. Esta distribución nos permitió trabajar de manera colaborativa, integrando nuestros conocimientos y habilidades en áreas específicas en las cuales se nos facilitaba trabajar, el producto final fue obtenido gracias a la responsabilidad, compromiso y trabajo de cada una de las áreas de trabajo del grupo.

El área de trabajo encargada de la parte de programación, integró los códigos en la plataforma de visual code, utilizando MongoDB y Mondoatlas para la construcción, desarrollo y visualización de las diferentes ventanas, así como la integración de servicios para los diferentes tipos de usuario del software. La plataforma GitHub fue



clave para el desarrollo colaborativo de nuestro proyecto, ya que nos proporcionó un repositorio compartido que sirvió como un área central de trabajo. GitHub no solo nos permitió almacenar y versionar nuestro código, sino que también facilitó la colaboración efectiva entre los miembros del equipo. Este curso marcó el cierre de todo el proceso de documentación, desarrollo e implementación del software educativo, consolidando las habilidades en frontend y en la estructuración de la interfaz de usuario, dando como resultado un software funcional, bien diseñado y con una interfaz intuitiva para los usuarios.

## Referencias

*Schwaber, K., & Sutherland, J. (2017). The Scrum Guide. Scrum.org.*

*Sutherland, J. (2014). Scrum: The Art of Doing Twice the Work in Half the Time. Crown Business.*