

Ingeniería De Software: Práctico N°1

- 1) Entre las estrategias que planteamos, una muy importante fue la división de tareas entre los integrantes del grupo. Algo que nos facilitó esto, y la organización del proyecto en sí, fue elegir una metodología ágil de trabajo. Al elegir SCRUM pudimos dar énfasis en la organización de bloques de trabajo, donde íbamos definiendo puntos esenciales.

Otra ventaja de SCRUM es el tema del control del trabajo y de la supervisión, que al realizar scrum meeting, con los scrum master (profes) se nos iba facilitando esto.

También al elegir un sistema de control de versiones, como git, era más fácil ir mejorando de a poco el software.

Además, algo que nos sirvió mucho fue construir un diagrama de objetos y un modelo de cómo queríamos que sea nuestra aplicación para tomarlo como base y poder guiarnos de él a medida que avanzábamos en el proyecto.

- 2) Para gestionar el personal, no tuvimos que esforzarnos mucho ya que éramos un equipo de solo 3 integrantes. Nos mantuvimos en contacto por redes (WhatsApp, Discord, etc.) y mediante reuniones que fuimos llevando a cabo en las horas libres en la universidad.

Para gestionar el proceso utilizamos git como control de versiones además de otras aplicaciones para realizar diagramas útiles como el diagrama de clases, el SRS, tablas con requerimientos funcionales y no funcionales, etc.

Finalmente, para gestionar problemas como por ejemplo: falta de tiempo para terminar una tarea planificada, problemas con la base de datos del sistema, poco conocimiento de las tecnologías a utilizar, etc. fuimos charlando entre nosotros a ver quién tenía más tiempo para tratar de terminar una tarea; buscar el problema por el cual falló la base de datos o el programa en sí; buscar y leer información o en internet sobre el funcionamiento de las tecnologías que debíamos utilizar.

- 3) Personas: El rol de cada una de los integrantes fue el mismo, podríamos indicar también como persona participante a los profesores, que se encargaron de ayudarnos en diferentes problemas y nos iban aconsejando y dando herramientas para poder realizar el software de calidad.

Procesos: Entre los procesos, primero definimos una plantilla, como SCRUM. Luego entre las herramientas que definimos, primero todas las tecnologías, frameworks y demás (Ruby, HTML, CSS, Sinatra, SQL), un sistema de control de versiones (git) entre otras herramientas.

Productos: Idea del producto, Captura de requisitos, análisis, diseño, implementación, prueba, planificación de tareas, control de versiones, etc.

- 4) Teniendo en cuenta que éramos un equipo de 3 integrantes, no había un líder, y la gestión del personal se realizaba entre todos, además de recibir motivación por parte del profesor del taller.

Si bien no teníamos un plan de gestión de riesgos bien definido, si tuvimos en cuenta los posibles fallos como por ejemplo: inconsistencias en la base de datos, errores en los modelos, errores en la gestión de métodos GET, POST, problemas con las sesiones, etc.

- 5) Si, ya que funcionamos muy bien como equipo de trabajo, cada uno aprendiendo del resto, compartiendo nuestros conocimientos apoyándonos mutuamente y resolviendo entre todos los integrantes los problemas que se nos iban presentando. Además, pudimos definir estándares de trabajo y tener una continua refactorización que nos permitio entregar un software de calidad.
- 6) Consideramos que si ya que desde un principio supimos tener una buena comunicación para poder dividirnos las tareas, ya que nos manteníamos en contacto de forma constante ya sea por Whatsapp, Discord, etc, lo cual nos ayudó a tener una mejor complementación y coordinación debido a que nos permitia trabajar en distintas áreas (cada integrante en una área distinta) para luego unificar y juntar todo. Por ejemplo, un integrante realizaba la base de datos, otro se encargaba de la semilla y el otro de lo visual.

Además, la división de tareas se hacía sabiendo o asegurando que todos los integrantes iban a estar comprometidos con su tarea, realizandola de forma correcta. Tomando el ejemplo anterior, el integrante que se encargaba de cargar las preguntas confiaba en que la base de datos iba a estar construida de forma correcta por otro integrante.