Retrospectiva

Este proyecto tuvo como objetivo implementar una red VPN funcional entre dispositivos y aplicar algoritmos de optimización (Dijkstra y Kruskal) para mejorar la transferencia de archivos y el diseño de la topología de red. Desde su concepción, se adoptó una metodología ágil basada en Scrum, lo que permitió organizar el trabajo en sprints cortos y fomentar la comunicación constante entre los miembros del equipo: David, Randu e Iván.

Durante la primera fase, se logró establecer la VPN entre los nodos, aunque no estuvo exenta de dificultades técnicas. El proceso de configuración implicó enfrentarse a problemas de compatibilidad entre dispositivos y redes locales, así como la necesidad de establecer reglas adecuadas de firewall y ruteo. Esta etapa resultó más demandante de lo previsto, lo que afectó el tiempo disponible para las fases posteriores.

En el segundo sprint, se implementaron los algoritmos de Dijkstra y Kruskal, con pruebas unitarias y simulaciones para verificar su funcionamiento. Aquí se evidenció la necesidad de integrar desde un inicio herramientas que permitieran automatizar las pruebas y validar el comportamiento de los algoritmos en escenarios reales. El trabajo colaborativo fue clave en esta etapa, y la definición previa de los roles ayudó a dividir eficientemente las tareas.

Finalmente, en el último sprint se realizaron las pruebas funcionales en la red real, integrando la lógica algorítmica con la VPN establecida. También se inició el proceso de documentación técnica. Aunque el resultado global fue exitoso, hubo presión de tiempo debido a que las etapas anteriores consumieron más recursos de los previstos.

A partir de esta experiencia, se identifican varias oportunidades de mejora para futuros proyectos:

1. **Planificación más realista de tiempos técnicos.** La configuración de redes, especialmente VPNs, implica variables impredecibles. Para mitigar esto, es recomendable destinar más tiempo a tareas de infraestructura y pruebas iniciales.

2. **Uso de entornos virtuales o simuladores en etapas tempranas.** Antes de desplegar en hardware real, puede usarse software como GNS3, Mininet o redes privadas virtuales simuladas, para evitar retrasos por limitaciones físicas.

3. **Incorporación de herramientas de automatización.** Scripts en Bash o Python para automatizar conexiones, ejecución de pruebas y análisis de rendimiento pueden ahorrar tiempo y reducir errores humanos.

4. **Sprint exclusivo para documentación.** Muchas veces, la documentación queda relegada al final del proyecto. Incluir un sprint dedicado exclusivamente a esto mejora la calidad del producto final.

5. **Retrospectivas intermedias.** Realizar mini retrospectivas después de cada sprint (además de la final) puede detectar cuellos de botella de manera más temprana.

En conclusión, este proyecto consolidó el conocimiento del equipo en redes, algoritmos y trabajo colaborativo bajo metodologías ágiles. Se lograron todos los OKRs planteados, y la experiencia dejó aprendizajes valiosos sobre gestión del tiempo, división de responsabilidades y adaptación frente a imprevistos técnicos. Estas lecciones servirán para mejorar futuros proyectos académicos o profesionales de índole similar.