



TECNOLÓGICO DE MONTERREY
Campus Monterrey

Análisis y diseño de algoritmos avanzados
(Gpo 602)

E2. Actividad Integradora 2 - Reflexion

<i>Fernando Rocha Martínez</i>		<i>A00836634</i>
<i>Daniel Gerardo Galvan Santacruz</i>		<i>A0128241</i>
<i>Ranferi Márquez Puig</i>		<i>A00832626</i>
<i>Manuel Adrián Gómez Castillo</i>		<i>A00836634</i>

Felipe Castillo Rendón
13 de mayo de 2025

REFLEXIÓN

Daniel Gerardo Galvan Santacruz:

Este proyecto me enseñó que programar no siempre es escribir código, sino construir una experiencia robusta y sostenible. Al principio, me enfoqué en que los algoritmos dieran el resultado correcto: Kruskal, TSP, Ford-Fulkerson... todos funcionaban. Pero pronto entendí que eso era solo la base. Lo más retador fue lograr que todo conviviera en un entorno profesional, donde cada cambio se prueba, se verifica y se analiza automáticamente.

Más allá de lo técnico, fue un ejercicio de disciplina: configurar CMake sin romper nada, entender cómo Google Test estructura las pruebas, o por qué SonarQube marcaba errores que parecían insignificantes. Aprendí a leer con atención, a probar antes de asumir, y a corregir detalles invisibles que marcan la diferencia entre un prototipo y un sistema profesional.

Este reto me cambió la manera de ver un proyecto: no basta con que funcione, tiene que escalar, mantenerse y cumplir estándares que otros puedan validar. Eso, para mí, es lo más cercano a trabajar como ingeniero en el mundo real.

Fernando Rocha Martínez:

Este proyecto comenzó como una implementación de algoritmos de grafos pero se convirtió en una lección profunda sobre desarrollo profesional moderno. Los algoritmos - Kruskal, TSP, Ford-Fulkerson y búsqueda geográfica - fueron relativamente directos de implementar. La verdadera complejidad surgió al integrar herramientas como CMake, Google Test, GitHub Actions y SonarQube en un ecosistema funcional. Cada herramienta tenía sus propias particularidades que debían trabajar en armonía, convirtiendo lo que parecía simple en un ejercicio complejo de ingeniería de sistemas. Problemas como formatos de archivos incompatibles o configuraciones conflictivas nos obligaron a profundizar en documentación técnica y entender los fundamentos de estas herramientas profesionales.

Creo que la lección más valiosa fue descubrir que las habilidades críticas no son algorítmicas sino de integración y operación de sistemas. Alcanzar 85%+ de cobertura de código y cero issues en SonarQube requirió entender qué significa "calidad" en términos medibles y construir infraestructura que la valide automáticamente. El pipeline resultante que ejecuta pruebas, genera reportes y valida calidad con cada commit representa exactamente lo que diferencia equipos maduros de amateur. Este proyecto cambió nuestra perspectiva sobre qué significa "terminar" software - no cuando los algoritmos funcionan, sino cuando tienes un sistema que valida automáticamente su propia calidad y garantiza estándares futuros. La capacidad de resolver problemas complejos de infraestructura y persistir através de errores técnicos oscuros son competencias que se transfieren a cualquier proyecto profesional.

Ranferi Márquez Puig:

Este proyecto me hizo pensar mucho sobre lo que realmente significa construir software. Claro, implementar algoritmos como Kruskal, TSP, Ford-Fulkerson y búsqueda

geográfica fue difícil, pero lo más complicado fue convertir esas funciones en un sistema sólido y fácil de mantener que cumpliera con los estándares profesionales.

Lo que más aprendí no fue la teoría algorítmica, sino lo que vino después: lidiar con el mundo real del desarrollo. Integrar herramientas para la construcción, pruebas, GitHub para automatización y SonarQube para análisis de calidad me mostró lo importante que es tener una infraestructura bien diseñada. Cada integración implicó tomar decisiones técnicas, resolver conflictos y leer mucha documentación, y no podría haberlo hecho sin la ayuda de mis compañeros.

Así que, para mí, un ingeniero profesional no solo resuelve problemas, sino que diseña entornos donde los errores no se repiten.

Manuel Adrián Gómez Castillo:

Para hacer este reto se ocupó aprender y entender los algoritmos avanzados, conforme avanzaba el semestre los ejercicios se hacían cada vez más complicados y complejos de entender así como también se ocupaba mucho el uso de la lógica y ver el más mínimo detalle para que pueda funcionar el código. Para el reto se ocupó mucho uso de lo visto en clase así como las técnicas aprendidas así como también saber cuando un código funciona como uno normal y cuando uno funciona de manera de que se distinga de todos los códigos por su infraestructura su entendimiento y la complejidad que este tiene