

REFLEXIÓN para fin de Unidad Temática 3

Las siguientes son algunas preguntas que pueden servirte como guía para tu reflexión personal sobre tus procesos de aprendizaje, utilizando como medio los objetivos y actividades de esta Unidad Temática.

No es obligatorio que las contestes todas (son sólo una guía!!), e incluso puedes (y se recomienda que lo hagas) explorar otras inquietudes que no estén aquí representadas: lo importante es que, por unos minutos, te concentres en **tus** procesos de aprendizaje y medites sobre ellos, con el objetivo de procurar identificar acciones que te permitan mejorarlos.

Recuerda referenciar todas las actividades que te han resultado beneficiosas, y en las carpetas correspondientes, poner las evidencias (trabajos extras, recursos adicionales consultados, ejercicios adicionales realizados y aprendizajes correspondientes obtenidos, etc.).

Adicionalmente se sugiere analizar las “war stories” recomendadas como anécdotas para la Unidad Temática y reflexionar sobre las mismas (puedes agregar un documento de reflexión específico, y cualesquiera otros artefactos que sean convenientes).

- ¿Cuáles son los resultados esperados del aprendizaje de esta Unidad Temática? (escribe lo que tú entiendas que son y significan).

Los resultados que esperaba para esta unidad eran poder entender el análisis de algoritmos y el uso adecuado de estructuras de datos como listas, pilas y colas. Independientemente de saber cómo funcionan técnicamente, quería entender cuándo y por qué usar una estructura u otra, y cómo eso afecta el rendimiento de los programas que hacemos, con esto también me propuse a escribir código más limpio y profesional, siguiendo los estándares de codificación en Java.

- ¿He alcanzado esos resultados? Documenta la(s) evidencia(s), si corresponde.

Sí, siento que logré varios de esos objetivos, más con los ejercicios grupales como el TA2 y el TA6, además de los de listas, tanto ArrayList como LinkedList, y es que con ellos aprendí cómo elegir la estructura adecuada según el contexto. También entendí mejor cómo funciona la eficiencia del código gracias a la notación O, permitiendo comparar algoritmos para ver de qué manera esas pequeñas decisiones pueden tener grandes impactos en el rendimiento.

Usar los estándares de codificación en los trabajos prácticos también me ayudó a mejorar la legibilidad del código, y considero que también fue una buena práctica.

- ¿Qué he aprendido? ¿Por qué aprendí?

Aprendí a mirar a los algoritmos desde un lado más analítico, y ver no solo si el código funciona, sino también en cómo funciona y qué tan eficiente es o puede llegar a ser. También aprendí sobre estructuras de datos abstractas, cómo se implementan en Java, y por qué es importante entender tanto la teoría como su implementación, permitiendo escribir código ordenado, con buenas prácticas, evitar errores y facilitando mucho el trabajo en equipo, ya que permitía de alguna manera tener un lenguaje universal de cómo programar.

- ¿Cuándo aprendí? ¿En qué circunstancias? ¿Bajo qué condiciones?

El aprendizaje se dio mayormente (medio como siempre) en los trabajos prácticos y un poco en la lectura del libro de Weiss, más específicamente en las secciones donde se comparan algoritmos, ya que como mencioné anteriormente, me ayudó a ver que el código que uno escribe tiene muchas formas de resolverse, pero no todas son igual de eficientes. También durante los ejercicios con listas fue donde más usé conceptos, porque me tocó usar la teoría en algo concreto.

- ¿Cómo he aprendido – o cómo no? ¿Sé qué tipo de aprendiz soy?

Confirmando que soy un aprendiz práctico, y que no puedo simplemente observar la masa de cocina y cómo se hace, sino que tengo que actuar, ya que me cuesta aprender solo con la teoría, por eso cuando empecé a aplicar lo que leía en ejercicios concretos, los conceptos empezaron a quedarse. También me ayuda mucho ver cómo lo resuelven otros y comparar soluciones, que claramente pasa mucho cuando trabajamos en grupo.

- ¿Cómo encaja lo que he aprendido en un plan completo y continuo de aprendizaje?

Esto encaja perfecto con el objetivo general del curso, que es desarrollar habilidades para programar algoritmos de forma correcta y eficiente. Ahora que tengo una base más sólida sobre cómo analizar algoritmos y qué estructuras de datos elegir, siento que estoy mejor preparado para enfrentar los temas que vienen. Además, seguir aprendiendo buenas prácticas de codificación me va a acompañar en cualquier proyecto futuro.

- ¿Qué diferencia ha producido el aprendizaje en mi desarrollo intelectual, personal y ético?

Intelectualmente, me siento más preparado para pensar en soluciones no solo funcionales, sino también optimizadas. Personalmente, creo que gané en organización mental, ya que empezar a pensar en términos de eficiencia te obliga a planear mejor. Y éticamente, reforcé la idea de que escribir código limpio y comprensible no es solo por uno mismo, sino por respeto al trabajo de los demás que puedan leerlo o mantenerlo.

- ¿Dónde, cuándo y cómo me he involucrado en aprendizaje integrado? ¿Ha sido mi aprendizaje conectado y coherente?

El aprendizaje fue integrado porque combinamos teoría (cosas como los capítulos del libro) con práctica (los TA y PD con listas, pilas y colas). El análisis de algoritmos se vio tanto en el papel como en el código real. También el uso de pseudocódigo me ayudó a planear mejor mis soluciones antes de escribirlas, haciendo con todo esto que el aprendizaje fuera bastante coherente y aplicado.

- ¿Es mi aprendizaje relevante, aplicable y práctico?

Sí, sin duda, ya que todo lo que vimos sobre estructuras de datos, análisis de eficiencia, y las convenciones de código se puede aplicar en cualquier proyecto, tanto académico como profesional. Sentí que esta unidad fue como un puente entre la programación básica y una programación más profesional y pensada.

- ¿Cuándo, cómo y por qué mi aprendizaje me ha sorprendido?

Me sorprendió ver cómo con un cambio en la estructura de datos, un algoritmo puede pasar de ser lento a súper rápido, por ejemplo, usar LinkedList en vez de ArrayList en ciertas situaciones hace una gran diferencia en las funcionalidades que puede tener el código o en la rapidez que puede llegar a tener. También me sorprendió ver que entender logaritmos y notación O no es tan complicado como pensaba, si se lo ve con ejemplos bien explicados (que igual costó encontrarlos).

- ¿De qué maneras mi aprendizaje ha sido valioso?

Este aprendizaje me dio herramientas para no programar a lo bruto, y la confianza de que lo que estoy programando es correcto, al no solo ver que funciona, si no que desde una visión más técnica y ordenada noto que una parte del código fue la mejor solución al problema. Siento que lo que aprendí no solo es útil ahora, sino que lo voy a usar en el resto de la carrera y en cualquier proyecto serio que encare.

- ¿Qué diferencia ha hecho, para mi aprendizaje, la tutoría sobre el portafolios?

La tutoría me sirvió como guía para no dejar todo en la cabeza y volcar las cosas por escrito, lo que ayuda mucho a ordenar ideas y ver realmente qué aprendí. Además, tener una estructura clara para reflexionar sobre lo aprendido me obliga a sentarme y tomarme un tiempo para ver el avance real que tuve.