



RETROSPECTIVAS

POOB



8 DE NOVIEMBRE DE 2025
ANDRES FELIPE PINEDA GAITAN
HEIDY ALEJANDRA ORJUELA RAMIREZ

- **CICLO 1**

1. ¿Cuáles fueron los mini-ciclos definidos? Justifiquenlos.

Durante el Ciclo 1 se definieron los siguientes mini-ciclos:

1. Diseño del simulador: Creación del diagrama de clases y secuencias en Astah, definiendo las clases principales y la estructura del proyecto.
2. Construcción de clases básicas: Implementación de la clase principal SilkRoad y sus métodos esenciales, respetando la herencia de los componentes del paquete shapes.
3. Implementación de funcionalidades básicas: Creación de métodos para añadir/eliminar tiendas y robots, mover robots, reabastecer tiendas y reiniciar la ruta de seda.

2. ¿Cuál es el estado actual del proyecto en términos de mini-ciclos? ¿Por qué?

El proyecto está completo en cuanto a las funcionalidades básicas definidas para este ciclo, pero presenta errores de lógica en algunos métodos, como la recolección de ganancias y el movimiento de robots en posiciones límite. Esto significa que los mini-ciclos de diseño y construcción fueron completados, pero todavía es necesario mirar los errores para asegurar que todo funcione correctamente.

3. ¿Cuál fue el tiempo total invertido por cada uno de ustedes? (Horas/Hombre)

- Heidy: 20 horas.
- Felipe: 20 horas.

4. ¿Cuál consideran fue el mayor logro? ¿Por qué?

El mayor logro fue completar todas las funcionalidades básicas del simulador, incluyendo la creación de la ruta de seda, la adición y eliminación de robots y tiendas, y la visualización del simulador.

5. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?

El mayor problema técnico fue la lógica de algunas funciones, especialmente el movimiento de robots y el cálculo de ganancias, que no siempre producían resultados correctos. Para resolverlo se realizaron pruebas revisando los métodos y ajustando la lógica en la implementación, aunque algunos errores persistieron y quedaron pendientes para los ciclos siguientes.

6. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?

Lo que hicimos bien:

- Planificación de mini-ciclos claros.

- Coordinación en el diseño y uso de Astah.
- Repartir las tareas para implementar funcionalidades de manera ordenada.

Compromisos de mejora:

- Revisar más detalladamente la lógica antes de finalizar los métodos.
- Realizar pruebas más exhaustivas para detectar errores tempranos.
- Documentar mejor.

7. Considerando las prácticas XP incluidas en los laboratorios, ¿cuál fue la más útil? ¿Por qué?

Durante este ciclo implementamos **Pair Programming**, donde uno de nosotros escribía el código mientras el otro revisaba y sugería mejoras.

8. ¿Qué referencias usaron? ¿Cuál fue la más útil?

- Documentación de Java y Javadoc para la correcta documentación de clases y métodos.
- Guías de BlueJ y el paquete Shapes para reutilización de componentes.
- Tutoriales de Astah para diseño de diagramas de clases y secuencia.

- **CICLO 2**

1. ¿Cuáles fueron los mini-ciclos definidos? Justifiquenlos.

Durante el Ciclo 2 se definieron los siguientes mini-ciclos:

1. Diseño y extensión del simulador: Actualización de los diagramas en Astah para incluir nuevas funcionalidades como movimientos automáticos de los robots y registro de ganancias.
2. Implementación de movimientos inteligentes de robots: Desarrollo de la lógica que permite a los robots decidir sus movimientos buscando maximizar la ganancia.
3. Implementación de funciones que registran cuántas veces se desocupa cada tienda y las ganancias obtenidas por cada robot en cada movimiento.
4. Pruebas de unidad y validación invisible

2. ¿Cuál es el estado actual del proyecto en términos de mini-ciclos? ¿Por qué?

El proyecto se encuentra completo en términos de construcción del simulador y de las nuevas funcionalidades del Ciclo 2, incluyendo:

- Movimientos automáticos de los robots que buscan maximizar la ganancia.
- Registro de desempeño de tiendas y robots.
- Pruebas de unidad ejecutadas en modo invisible.

En este ciclo teníamos solo la parte lógica del proyecto, nos falta la implementación en el canva.

3. ¿Cuál fue el tiempo total invertido por cada uno de ustedes? (Horas/Hombre)

- Heidy: 22 horas.
- Felipe: 22 horas.

4. ¿Cuál consideran fue el mayor logro? ¿Por qué?

El mayor logro fue implementar la lógica de movimiento automático de los robots para maximizar la ganancia, junto con el registro de desempeño de cada robot y tienda..

5. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?

El mayor problema técnico fue diseñar la lógica que permitiera a los robots tomar decisiones inteligentes sin que el programa se volviera demasiado lento o complicado. Se resolvió mediante:

- Implementación paso a paso de la lógica de decisión.
- Pruebas unitarias para verificar cada escenario.

6. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?

Lo que hicimos bien:

- Coordinación en la creación de pruebas unitarias colectivas.
- Distribución de tareas claras: diseño, programación y pruebas.
- Documentación de los casos de prueba y de las decisiones de diseño.

Compromisos de mejora:

- Mejorar la planificación de mini-ciclos para incluir optimización de la lógica desde el inicio.
- Rotar los roles en Pair Programming para que todos tengan experiencia codificando y revisando.
- Incluir más escenarios en pruebas unitarias para cubrir casos límite.

7. Considerando las prácticas XP incluidas en los laboratorios, ¿cuál fue la más útil? ¿Por qué?

La práctica más útil fue Pair Programming, ya que uno codifica mientras otro revisa el código y sugiere mejoras. Esto ayudó a:

- Detectar errores de inmediato durante la implementación de la lógica de movimiento de los robots.
- Mejorar la calidad del código desde el principio.
- Compartir conocimiento del proyecto entre todos los miembros del equipo.

8. ¿Qué referencias usaron? ¿Cuál fue la más útil?

- Documentación de Java y Javadoc para la correcta implementación y documentación de clases y métodos.
- Tutoriales y ejemplos de lógica de decisiones y algoritmos de optimización para los movimientos de robots.

CICLO 3

1. ¿Cuáles fueron los mini-ciclos definidos? Justifíquenlos.

Durante el Ciclo 3 se definieron los siguientes mini-ciclos:

- Mini-ciclo de diseño: Se realizó en Astah para definir la estructura de clases, relaciones y diagramas necesarios para el simulador y la solución del problema de la maratón.
- Mini-ciclo de implementación: Se desarrolló el código Java siguiendo los estándares, implementando las clases del simulador y de solución del problema.
- Mini-ciclo de pruebas: Se crearon pruebas unitarias (SilkRoadContestTest) y de casos comunes (SilkRoadContestCTest) para verificar la funcionalidad de cada componente.
- Mini-ciclo de integración y corrección: Se integraron las diferentes partes y se realizaron ajustes debido a problemas de eficiencia y manejo de estructuras (por ejemplo, uso de arreglos en lugar de ArrayList).

2. ¿Cuál es el estado actual del proyecto en términos de mini-ciclos? ¿por qué?

El proyecto se encuentra completo en funcionalidad básica:

- La solución del problema de la maratón se implementó correctamente.
- El simulador está operativo, pero el GUI solo contiene los botones principales y no todos los elementos gráficos deseados.
- Se realizaron ajustes de estructura, pero aún se usan arreglos en lugar de ArrayList, lo que afecta la eficiencia y extensibilidad.

3. ¿Cuál fue el tiempo total invertido por cada uno de ustedes? (Horas/Hombre)

- Heidy: 17 horas.
- Felipe: 17 horas.

4. ¿Cuál consideran fue el mayor logro? ¿Por qué?

El mayor logro fue implementar correctamente la solución del problema y simularla en el sistema. Esto demuestra que comprendemos la lógica del problema en un código funcional y probarlo con diferentes valores.

5. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?

El mayor problema técnico fue la eficiencia y estructura de datos, ya que se usaban arreglos en lugar de ArrayList, lo que limitaba la extensibilidad y facilidad de manejo de los robots y tiendas. Para resolverlo:

- Se ajustaron las clases para que funcionaran de la manera correcta y con las recomendaciones dadas.

6. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?

Bien como equipo:

- Buena división de tareas según habilidades.
- Coordinación efectiva en el desarrollo y las pruebas unitarias.
- Documentación clara del diseño en Astah.

Compromiso para mejorar:

- Optimizar estructuras de datos (ArrayList, colecciones de Java).
- Mejorar la interfaz gráfica agregando todos los botones y funcionalidades faltantes.
- Realizar pruebas de integración más exhaustivas para evitar ajustes de última hora.

7. Considerando las prácticas XP incluidas en los laboratorios. ¿cuál fue la más útil? ¿por qué?

La práctica más útil fue TDD (Desarrollo guiado por pruebas), ya que permitió:

- Detectar errores de lógica desde etapas tempranas.
- Garantizar que los métodos implementados cumplieran con los requisitos.
- Facilitar la corrección de fallos en iteraciones posteriores.

8. ¿Qué referencias usaron? ¿Cuál fue la más útil? Incluyan citas con estándares adecuados.

Referencias utilizadas:

- Oracle Java Documentation. Oracle. Disponible en: <https://docs.oracle.com/javase/>
- Maratón de Programación ICPC 2024 – Problem J “The Silk Road ... with Robots!”. Disponible en: <https://icpc.global/>
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

CICLO 4

1. ¿Cuáles fueron los mini-ciclos definidos? Justifíquenlos.

Durante el Ciclo 4 se definieron los siguientes mini-ciclos:

- Implementación del simulador básico con robots y tiendas normales.
- Añadir funcionalidades de reposición de tiendas y reinicio de robots, además de pruebas de unidad iniciales.
- Incorporación de nuevos tipos de robots y tiendas, refinamiento de la lógica de movimiento y captura de dinero.
- Refactorización y extensión del sistema (silkRoad), implementación de todos los tipos de robots y tiendas, pruebas de aceptación y entrega de documentación completa.

2. ¿Cuál es el estado actual del proyecto en términos de mini-ciclos? ¿por qué?

El proyecto se encuentra completando el cuarto ciclo, que representa la fase final de refinamiento y entrega. Esto se debe a que ya se han implementado los tipos de robots y tiendas, se realizó la refactorización de los paquetes, se crearon pruebas de unidad y aceptación, y se está finalizando la documentación y la retrospectiva del proyecto.

3. ¿Cuál fue el tiempo total invertido por cada uno de ustedes? (Horas/Hombre)

- Heidy: 25 horas.
- Felipe: 25 horas.

4. ¿Cuál consideran fue el mayor logro? ¿Por qué?

El mayor logro fue la implementación de diferentes tipos de robots y tiendas con comportamientos diferenciados, junto con la refactorización de los paquetes usando herencia. Esto permitió un sistema más extensible, modular y fácil de mantener.

5. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?

El mayor problema técnico fue coordinar los comportamientos distintos de robots y tiendas sin que se generaran conflictos en la lógica de interacción

Se resolvió mediante:

- Refactorización de las clases usando herencia y polimorfismo, permitiendo que cada tipo sobrescriba el comportamiento específico.
- Creación de pruebas unitarias para validar cada comportamiento antes de integrarlo en el simulador.

6. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?

- Lo que hicieron bien: coordinación en la creación de pruebas de aceptación y unidad, uso de Astah para diseño y documentación clara del código siguiendo estándares Java, división de tareas por paquetes y tipos de robots/tiendas.
- Compromiso de mejora: mejorar la planificación de tiempos y la comunicación durante la refactorización, así como documentar de manera más detallada los resultados de pruebas para futuras entregas.

7. Considerando las prácticas XP incluidas en los laboratorios, ¿cuál fue la más útil? ¿por qué?

La práctica más útil fue las pruebas frecuentes de aceptación y unidad, porque permitieron detectar errores tempranamente y asegurar que las nuevas funcionalidades no rompieran las existentes, facilitando la integración continua y el desarrollo iterativo del simulador.

8. ¿Qué referencias usaron? ¿Cuál fue la más útil? Incluyan citas con estándares adecuados.

- Documentación oficial de Java: Oracle Java SE Documentation. <https://docs.oracle.com/en/java/javase/>
- Artículos sobre principios SOLID y refactorización: Martin, R. (2003). *Agile Software Development, Principles, Patterns, and Practices*. Prentice Hall.