

**DESARROLLO ORIENTADO A OBJETOS**

**Construcción. Clases y objetos.**

**Retrospectiva proyecto inicial, ciclo 1**

**2025-2**

**Mateo Alejandro Sánchez Buitrago**

**Camilo Alejandro Aguirre**

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

Se definieron tres mini-ciclos, fueron los siguientes:

1.a. El diseño inicial: se hizo la construcción del diagrama de clases y se definió responsabilidades en cada clase.

1.b. La implementación básica: Se hizo la creación de las clases principales, que eran: SilkRoadSimulator, SilkRoadModel, SilkRoadView, Robot, Store. Además, se hizo la validación de compilación.

1.c. La integración con shapes y visualización: Se incorporo el paquete shapes para la representación gráfica. Se ajustaron los métodos: eraseAll, getStores, getRobots. Y se hizo la verificación en BlueJ.

Estos mini-ciclos fueron definidos para avanzar del diseño a lo concreto que es la implementación y visualización, asegurando que haya correcciones rápidas en cada etapa.

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

El proyecto se encuentra en el tercer mini-ciclo concluido y logramos que: El código está completo y compile sin errores. Se integró con shapes para cumplir con el requisito de la visualización. Se realizaron los diagramas UML (clases y secuencia). Se alcanzó el objetivo del ciclo final.

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

Integrante Camilo Aguirre: 10 horas.

Integrante Mateo Sanchez: 8 horas.

Total: 18 horas en total.

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

El mayor de los logros fue lograr la integración correcta del paquete shapes en BlueJ. Permitió cumplir con el requerimiento gráfico, mostró la dinámica y tiendas en el modelo. Este paso aseguró que el sistema no fuera solamente lógico, sino también visual.

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

El mayor problema fue no lograr la compatibilidad con métodos que son inexistentes en shapes como: eraseAll, getStores, getRobots.

Para solucionarlo se editaron los códigos de SilkRoadView y SilkRoadModel. Se justaron los métodos a la versión del paquete shapes. Esto se resolvió revisando las clases disponibles, las cuales eran: Circle, Rectangle, Canvas y logrando adaptar la lógica de dibujo.

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

Lo que hicimos bien: Mantener comunicación constante, nos conocemos de hace buen tiempo entonces además de una comunicación constante se logra tener una buena comunicación. Tambien se dividieron algunas partes y se verificaron avances en cada paso.

Compromiso: Tratar de usar el tiempo desde que habilitan y nos informan de la tarea. Asi no tener presura a ultimo momento. Además, planificar una estrategia al inicio para llevar un orden lógico.

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

La práctica más útil fue el diseño incremental en los mini-ciclos. Esto permitió ir corrigiendo problemas a tiempo sin afectar todo el proyecto despues de un gran avance. Aplicando esto los errores de integración con shapes se resolvieron sin rehacer por completo el trabajo o afectar algunas partes por corregir otras.

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

Kolling, M., & Barnes, D. (2016). Objects First with Java: A Practical Introduction using BlueJ. Pearson.

Documentación de BlueJ. (2025). Disponible en: [https://bluej.org](https://bluej.org)

<https://www.youtube.com/watch?v=t8kR7urUTu>

<https://www.youtube.com/watch?v=KJd1Un8SEnk>

<https://www.youtube.com/watch?v=Z0yLerU0g-Q>

<https://www.youtube.com/watch?v=-jZ8JkC-Pwc>

<https://www.youtube.com/watch?v=t8kR7urUTuA&t=844s>

Funcionalidades pendientes. Terminamos el proyecto con todo lo básico funcionando y cumpliendo lo que se pidió en cuanto al diseño. Sin embargo, quedaron pendientes algunas mejoras para más adelante:

* Espiral cuadrada: Por ahora, robots y tiendas se muestran en línea recta, pero la idea original era ubicarlos en forma de espiral. Eso requiere cálculos más complejos, así que lo dejamos para después.
* Barra de ganancias: Las ganancias ya se calculan, pero no logramos mostrar una barra gráfica para que representarlas.
* Mensajes de error: Si algo no se puede hacer (como usar un robot inexistente), no dice nada. Nuestra idea es agregar mensajes que expliquen lo que ocurrió.

Nos enfocarnos en que todo lo esencial funcionara bien, y dejamos estas mejoras para futuras versiones.