

ESCUELA COLOMBIANA DE INGENIERÍA

DESARROLLO ORIENTADO POR OBJETOS [DOPO-POOB]

PROYECTO INICIAL Ciclo No. 2 2025-2

REFACTORIZING Y EXTENSIÓN

El proyecto inicial tiene como propósito desarrollar una aplicación que permita simular una situación inspirada en el **Problem J** de la maratón de programación internacional 2024 **The Silk Road ... with Robots!**. En este simulador se desea poder experimentar de forma más libre con la situación de la ruta de seda.

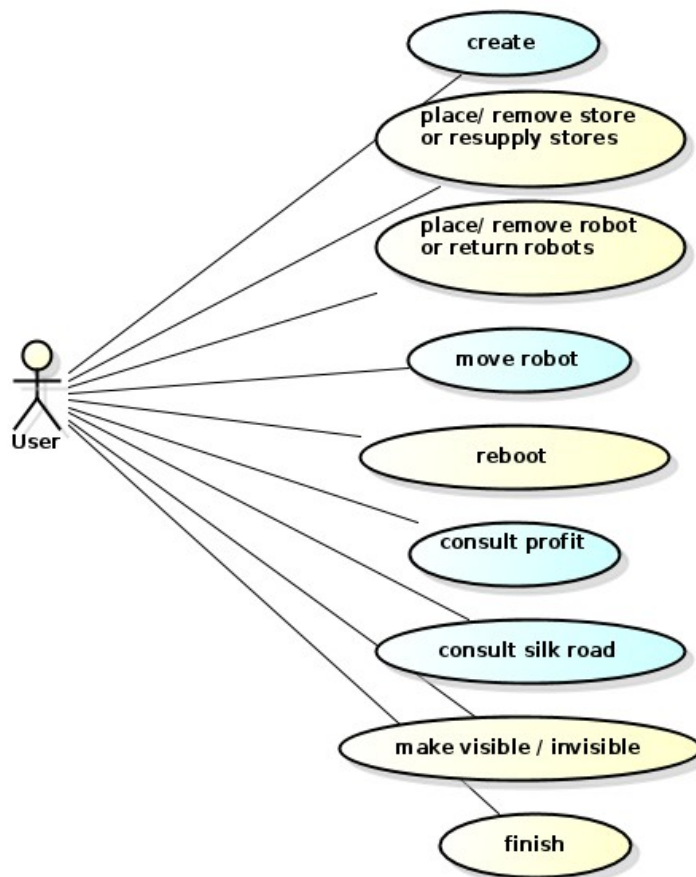
SEGUNDO CICLO

Los requisitos para el primer ciclo de desarrollo están indicados a continuación. Siempre hay un requisito implícito: el de **EXTENSIBILIDAD**.

MUY IMPORTANTE PLANIFICAR LOS MINI-CICLOS PARA ORIENTAR EL DESARROLLO
En esta entrega NO deben resolver el problema de la maratón sólo deben construir el simulador.

REQUISITOS FUNCIONALES

10. Debe permitir crear una ruta de seda con la entrada del problema de la maratón¹
11. Debe permitir a los robots decidir sus movimientos buscando maximizar la ganancia
12. Debe permitir consultar el número de veces en que cada tienda ha sido desocupada
13. Debe permitir consultar las ganancias que ha logrado cada robot en cada movimiento



- **create** Extensión.
Corresponde a requisito 10.
- **move robot.** Extensión.
Corresponde a requisito 11.
- **consult.** Extensión.
Corresponde a los requisitos 12 y 13.

¹ El número de días no se incluye, depende de la longitud del vector. La acción reboot marca un nuevo día.

REQUISITOS DE DISEÑO Y CONSTRUCCIÓN

SilkRoad
<pre>+ _ (length : int) : SilkRoad + _ (days : int[][]) : SilkRoad + placeStore(location : int, tenges : int) : void + removeStore(location : int) : void + placeRobot(location : int) : void + removeRobot(location : int) : void + moveRobots() : void + moveRobot(location : int, meters : int) : void + resupplyStores() : void + returnRobots() : void + reboot() : void + porfit() : int + stores() : int[][] + emptiedStores() : int[][] + robots() : int[][] + profitPerMove() : int[][] + makeVisible() : void + makeInvisible() : void + finish() : void + ok() : boolean</pre>

emptiedStores: ordenadas por localización de menor a mayor. [location, times]
profitPerMove: ordenados por localización de menor a mayor. [location, profit_move_1, profit_move_2]

REQUISITOS DE USABILIDAD

- Las tiendas desocupadas deben lucir diferentes
- El robot que ha logrado mayor ganancia debe parpadear

REQUISITOS DE ENTREGA

Los productos esperados para esta entrega son:

1. Diseño completo en la herramienta astah
2. Implementación siguiendo los estándares de codificación y documentación de java.
3. Casos de pruebas de unidad de su proyecto: **SilkRoadc2Test**
Las pruebas de unidad deben ser en modo invisible.
No olviden diseñar las pruebas considerando dos preguntas: ¿qué debería hacer? ¿qué no debería hacer?
4. Propuesta de casos de prueba para incluir en la clase **SilkRoadcc2Test**
Las clase **SilkRoadcc2Test** será una creación colectiva usando el wiki correspondiente.
Los nombres de los casos de prueba deberán incluir la identificación de los autores. Por ejemplo, **accordingDAShould....** (DA: Iniciales de los primeros apellidos en orden alfabético).
5. Documento de retrospectiva del proyecto. (Ver ciclo uno)
Es necesario incluir la retrospectiva de este ciclo y de los anteriores.
6. No olviden preparar dos pruebas de aceptación para la presentación

Los productos los deben publicar en un repositorio Git y deben publicar en moodle un archivo txt que contenga la dirección URL. El nombre del archivo txt debe ser la concatenación de los apellidos de los autores, ordenados alfabéticamente.

DEFINIR EL ESTADO EN TÉRMINOS DE LOS MINI-CICLOS PLANIFICADOS

Publicar productos a revisión : Semana 06 21 de septiembre