

# ESCUELA COLOMBIANA DE INGENIERÍA

## PROGRAMACIÓN ORIENTADA A OBJETOS

### PROYECTO INICIAL Ciclo No. 2 2024-1

#### REFACTORIZING Y EXTENSIÓN

El proyecto inicial tiene como propósito desarrollar una aplicación que permita simular una situación inspirada en el **Problem I** de la maratón de programación internacional 2021 **Spider Walk**

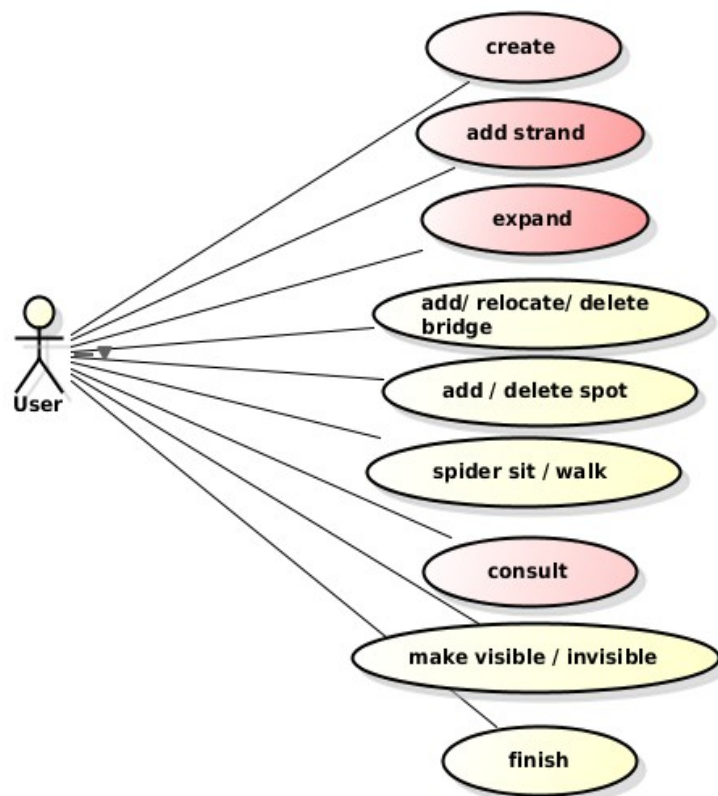
#### SEGUNDO CICLO

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

8. Debe permitir crear un Spiderweb con la entrada del Problem I.<sup>1</sup>
9. Debe permitir adicionar una hebra
10. Debe ampliar la telaraña
11. Debe permitir consultar los sitios favoritos a los que la araña puede llegar
12. Debe permitir consultar los puentes que no han sido usados



- **create** Extensión. Corresponde a requisito 8.
- **add strand**. Nuevo. Corresponde a requisito 9.
- **expand**. Nuevo. Corresponde a requisito 10
- **consult**. Extensión. Corresponde a los requisitos 11 y 12.

<sup>1</sup>El detalle de la entrada está en la sección Input del problema de la maratón.

## REQUISITOS DE DISEÑO Y CONSTRUCCIÓN

---

| Spiderweb   |
|---|
| <pre>+ _(strands : int, radio : int) : Spiderweb + _(strands : int, favorite : int, bridges : [][]int) : void + addStrand() : void + enlarge(percentage : int) : void + addBridge(color : String, distance : int, firstStrand : int) : void + relocateBridge(color : String, distance : int) : void + delBridge(color : String) : void + addSpot(color : String, strand : int) : void + delSpot(color : String) : void + spiderSit(strand : int) : void + spiderWalk(advance : boolean) : void + spiderLastPath() : int[] + bridges() : String[] + bridge(color : String) : int[] + spots() : String[] + spot(color : String) : int + reachableSpots() : String[] + unusedBridges() : String[] + makeVisible() : void + makeInvisible() : void + finish() : void + ok() : boolean</pre> |

Los colores deben retornarse en minúscula y ordenados alfabéticamente.

## REQUISITOS DE USABILIDAD

---

Los puentes usados deben lucir de forma diferente-

## 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: **Spiderwebcc2Test**  
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 **Spiderwebcc2Test**  
La clase **Spiderwebcc2Test** 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 el espacio preparado en moodle en un archivo .zip con un nombre igual a la concatenación de los apellidos de los autores, ordenados alfabéticamente.

**MUY IMPORTANTE DEFINIR EL ESTADO EN TÉRMINOS DE LOS MINI-CICLOS PLANIFICADOS**

Publicar productos a revisión

:

Semana 06 2 de marzo