

# ESCUELA COLOMBIANA DE INGENIERÍA

## PROGRAMACIÓN ORIENTADA A OBJETOS

### PROYECTO INICIAL Ciclo No. 2 2020-2

#### REFACTORIZING Y EXTENSIÓN

El proyecto inicial tiene como propósito desarrollar una aplicación que permita simular una situación inspirada en el **Problem E** de la maratón de programación internacional 2019 **Dead End Detector**. En esta versión vamos a identificar las ciudades por colores.

#### PRIMER CICLO

---

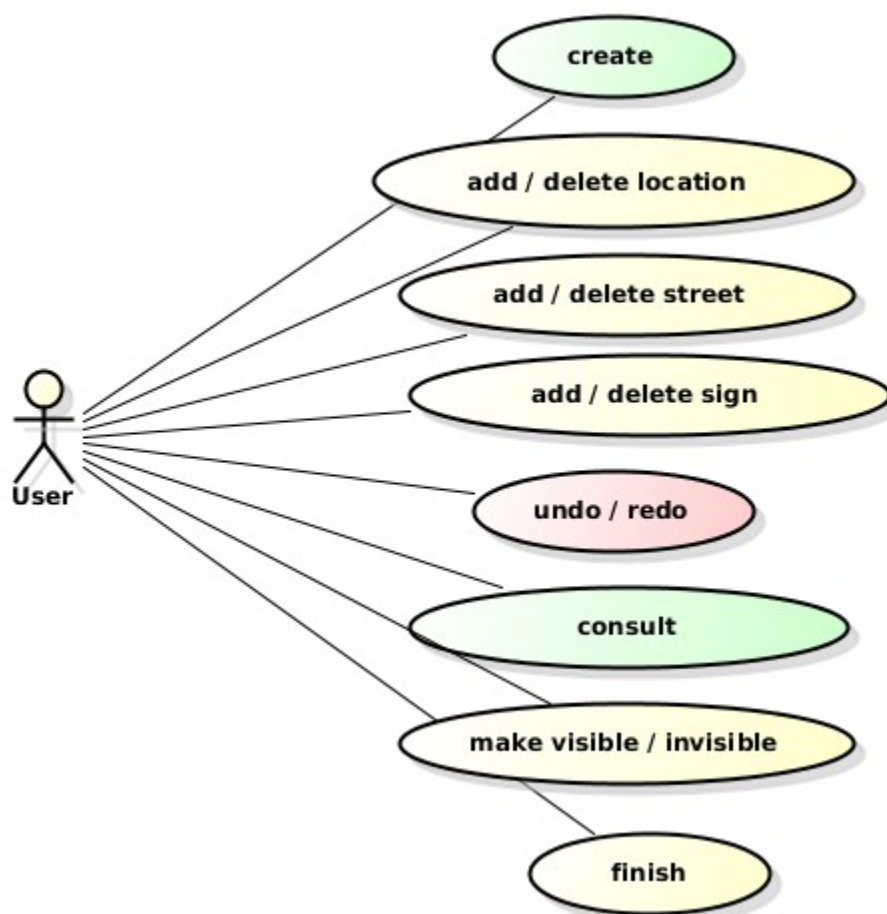
Los requisitos para el primer ciclo de desarrollo están indicados a continuación.  
No olviden que siempre hay un requisito implícito: el de **EXTENSIBILIDAD**.

**En esta entrega NO deben resolver el problema de la maratón sólo deben construir el simulador .**

#### REQUISITOS FUNCIONALES

---

8. Debe permitir crear una ciudad al azar dadas sus características
9. Debe permitir crear una ciudad dada la entrada definida en la maratón
10. Debe permitir deshacer y rehacer las acciones del simulador
11. Debe permitir consultar todos los elementos del simulador



- **create** Extensión.  
Corresponde al requisito 8,9
- **undo/redo.** Nuevo,  
Corresponde al requisito 10
- **consult** Extensión,  
Corresponde al requisito 1

## REQUISITOS DE DISEÑO Y CONSTRUCCIÓN

Town
<pre>+ _ (length : int, width : int) : Town + _ (length : int, width : int, locations : int, streets : int, signs : int) : Town + _ (input : String[]) : Town + addLocation(color : String, x : int, y : int) : void + addStreet(locationA : String, locationB : String) : void + addSign(locationA : String, locationB : String) : void + delLocation(color : String) : void + delStreet(locationA : String, locationB : String) : void + delStreet() : void + delSign(locationA : String, locationB : String) : void + delSign() : void + redo() : void + undo() : void + roadMap() : String[][] + allLocations() : String[][] + allStreets() : String[][] + unnecessaryStreets() : String[][] + allSigns() : String[][] + wrongSigns() : String[][] + makeVisible() : void + makeInvisible() : void + finish() : void + ok() : boolean</pre>

En el desarrollo deben respetar las decisiones de diseño presentes en este diagrama.

*El input del tercer creador sigue las especificaciones del enunciado de la maratón. Los colores asignados deben seguir el orden alfabético definido por los números.*

*El retorno de todas las consultas debe seguir el orden alfabético y debe estar en minúsculas.*

*El retorno de la consulta de **localidades** debe tener el color de la ciudad y su posición en x y en y.*

*El retorno de la consulta de calles debe tener los pares de localidades ordenados alfabéticamente.*

Los productos esperados para esta entrega son:

1. Diseño completo en la herramienta astah  
**Diagrama de clases** con atributos y métodos privados y públicos.  
**En astah, crear un diagrama de clases (dejar el nombre por omisión)**  
(De las clases de **shapes** incluya únicamente el detalle (atributos - métodos) de lo que extendieron  
**Diagrama de secuencia** de cada uno de los métodos de la clase **Town**  
**En astah, crear los diagramas de secuencia en los métodos (dejar el nombre por omisión)**  
(Parar en los componentes de **shapes**)
2. Código siguiendo los estándares de documentación de java.  
**No olviden que el código de los métodos no debe ocupar más de una pantalla.**
3. Propuesta de casos de prueba para clase **TownTestC2**.  
Las pruebas se deben preparar en modo invisible.  
No olviden diseñar las pruebas considerando dos preguntas: ¿qué debería hacer? ¿qué no debería hacer?  
La clase **TownTestC2** será una creación colectiva usando el wiki correspondiente.  
Los nombres de los casos de prueba deberán incluir la identificación de los autores. Por ejemplo, **segunDA** debería (DA: Iniciales de los primeros apellidos en orden alfabético).
4. Documento de retrospectiva del proyecto. (Ver ciclo uno)  
**Es necesario incluir la retrospectiva de este ciclo y de los anteriores.**
5. No olviden preparar dos pruebas de aceptación para la presentación

## REQUISITOS DE ENTREGA

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.

**Es necesario incluir la retrospectiva de este ciclo y de los anteriores.**

Publicar productos a revisión : Semana 06 10 de septiembre