

ESCUELA COLOMBIANA DE INGENIERÍA

PROGRAMACIÓN ORIENTADA A OBJETOS

PROYECTO INICIAL Ciclo No 1 2020-02

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

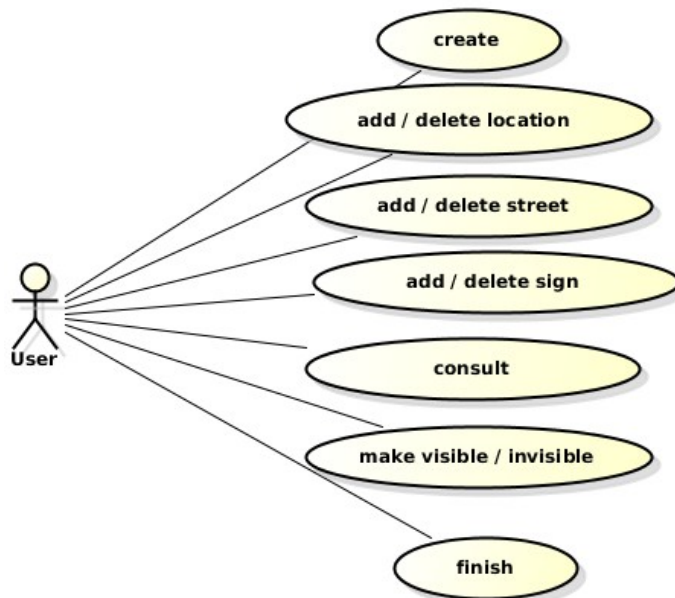
Los requisitos para el primer ciclo de desarrollo están indicados a continuación. No olviden que siempre hay un requisito implícito: el de **EXTENSIBILIDAD**.

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

REQUISITOS FUNCIONALES

El simulador debe permitir:

1. Crear una ciudad
2. Adicionar y eliminar una localidad
3. Adicionar y eliminar una calle
4. Adicionar y eliminar una señal
5. Consultar la información del mapa
6. Hacer visible o invisible el simulador
7. Terminar el simulador



create. Requisito 1.

add / delete location. Requisito 2.

add / delete street. Requisito 3.

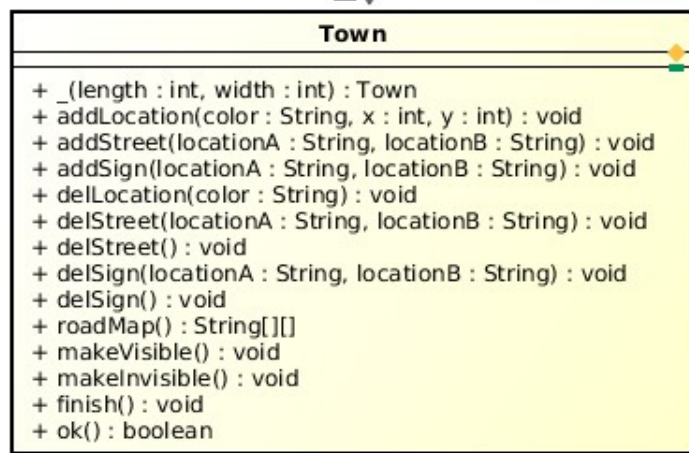
add / delete signal. Requisito 4.

consult Requisito 5.

make visible / invisible. Requisito 6.

finish. Requisito 7 .

REQUISITOS DE DISEÑO Y CONSTRUCCIÓN



REQUISITOS DE USABILIDAD

1. Las localizaciones deben ser de colores diferentes. Deben permitir visualizar mínimo 50 colores.
2. Los últimos elementos adicionados o eliminados deben distinguirse de los otros
3. Los elementos que se pueden eliminar deben también distinguirse
4. Si la acción no se puede realizar se le debe presentar un mensaje especial, sólo si el simulador es visible.
5. Las posiciones se dan en coordenadas cartesianas cuadrante positivo

REQUISITOS DE DISEÑO Y CONSTRUCCIÓN

1. En su desarrollo debe respetar las decisiones de diseño presentes en este diagrama de clases para la clase principal.
El método **ok** retorna si la última acción se pudo realizar o no.
2. Las clases se deben construir reutilizando los componentes del proyecto [shapes](#) que sean necesarios.
3. El paquete [shapes](#) puede ser extendido, si se requieren otras funcionalidades. Incluyan en la retrospectiva las extensiones y su justificación.
4. Las clases deben tener la documentación estándar de java. No olvidar revisar la documentación generada.
5. Las clases se deben construir en **BlueJ**. El nombre del nuevo proyecto debe ser **checkers**

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.

1. ¿Cuáles fueron los mini-ciclos definidos? Justifiquenlos.
2. ¿Cuál es el estado actual del laboratorio en términos de mini-ciclos? ¿por qué?
3. ¿Cuál fue el tiempo total invertido por cada uno de ustedes? (Horas/Hombre)
4. ¿Cuál consideran fue el mayor logro? ¿Por qué?
5. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?
6. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?
7. Considerando las prácticas XP del laboratorio. ¿cuál fue la más útil? ¿por qué?

Publicar productos a revisión : Jueves 27 de agosto