

# ESCUELA COLOMBIANA DE INGENIERÍA

## PROGRAMACIÓN ORIENTADA A OBJETOS

### PROYECTO INICIAL Ciclo No 1 2019-02

El proyecto inicial tiene como propósito desarrollar una aplicación que permita simular una situación inspirada en planteada en el **Problema F** de la maratón de programación internacional 2019 **Directing Rainfall**. En esta versión pueden existir varios viñedos.

#### 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**.

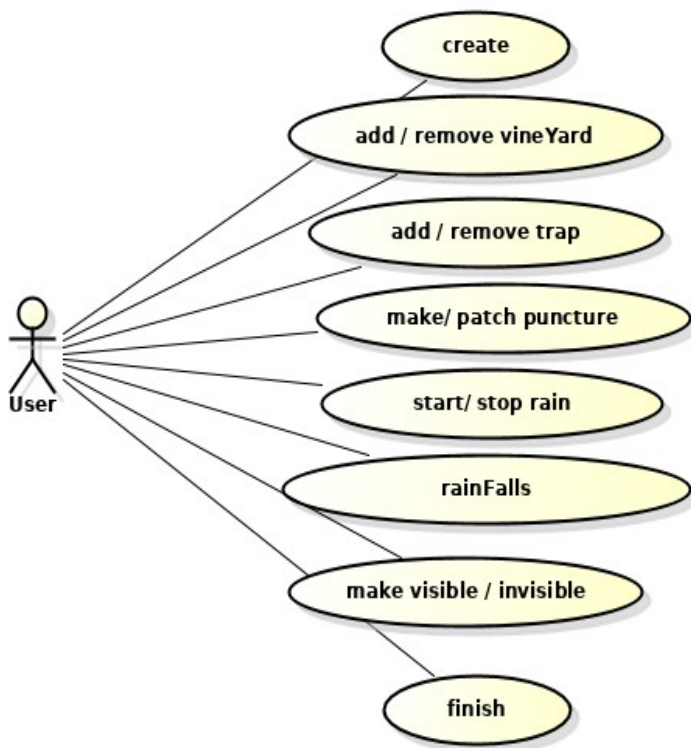
**En esta entrega NO deben resolver el problema de la maratón sólo deben construir el simulador .**

#### REQUISITOS FUNCIONALES

---

El simulador debe permitir:

1. Crear un valle
2. Adicionar y eliminar un viñedo
3. Adicionar, agujerear y eliminar una lona
4. Indicar cuando llueve y deja de llover
5. Consultar los viñedos que están recibiendo lluvia
6. Hacer visible o invisible el simulador (debe poder funcionar invisible)
7. Terminar el simulador



**create.** Requisito 1.

**add / remove vineyard.** Requisito 2.

**add / remove trap.** Requisito 3.

**make / patch puncture.** Requisito 3.

**start / stop rain.** Requisito 4.

**rainFalls.** Requisito 5.

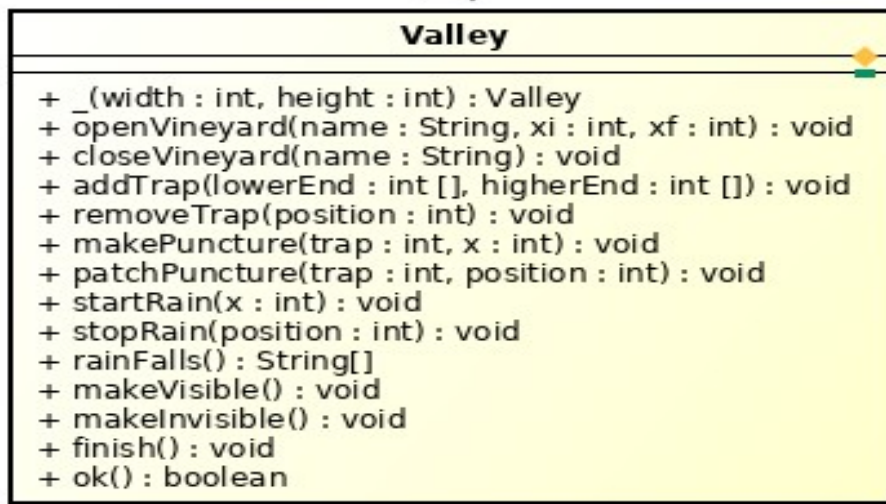
**make visible / invisible.** Requisito 6.

**finish.** Requisito 7 .

---

## REQUISITOS DE DISEÑO Y CONSTRUCCIÓN

---



## REQUISITOS DE USABILIDAD

---

1. Los viñedos se deben dibujar de tal forma que se distingan entre ellos.
2. Si las lonas cubren un sólo viñedo, deben ser del color del viñedo al que pertenecen. Sino deben ser negras.
3. Los agujeros de las lonas y los rayos de lluvia se deben ver claramente.
4. El simulador debe poder funcionar en modo visible o invisible, al iniciar es invisible
5. Si el usuario comete algún error se le debe presentar un mensaje especial, sólo si el simulador es visible.
6. Las posiciones se cuentan desde 1 de izquierda a derecha y de arriba a abajo.

## 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 operación 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 **valley**

## 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? Justifíquenlos.
2. ¿Cuál es el estado actual del laboratorio en términos de mini-ciclos? ¿por qué?
3. ¿Cuál fue el tiempo total invertido en el laboratorio 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 29 de agosto