

# ESCUELA COLOMBIANA DE INGENIERIA

## PROGRAMACIÓN ORIENTADA A OBJETOS

### PROYECTO INICIAL Ciclo No 1 2015-2

#### PROYECTO INICIAL

---

El proyecto inicial tiene como propósito desarrollar una aplicación que permita simular la situación planteada en el **Problema E** de la maratón de programación internacional 2015 **Evolution in Parallel**

#### 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**.

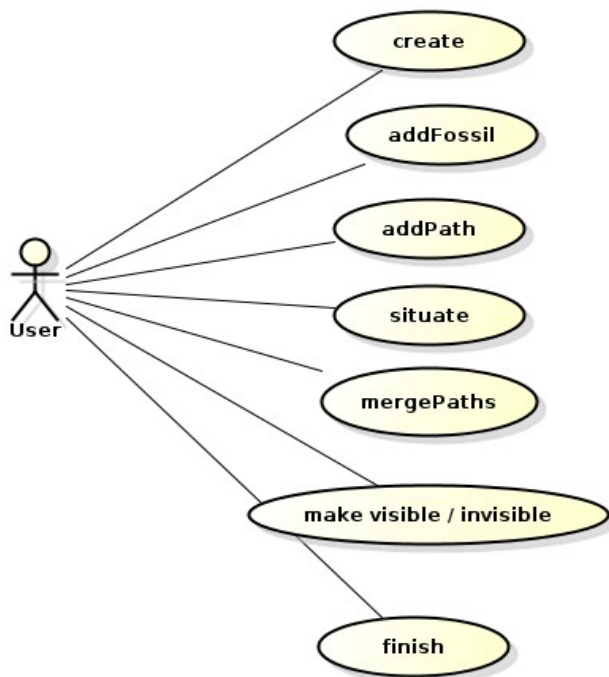
En esta entrega NO deben resolver el problema de la maratón sólo deben construir el simulador .

#### REQUISITOS FUNCIONALES

---

El simulador debe permitir:

1. Crear un planeta (dada la estructura genética de sus organismos)
2. Incluir un nuevo fósil (dada su estructura genética)
3. Adicionar una posible línea de evolución (pueden existir más de dos)
4. Ubicar un fósil en una línea de evolución (debe cumplir las reglas de evolución)
5. Mezclar dos líneas de evolución (debe cumplir las reglas de evolución)
6. Hacer visible o invisible el simulador (debe poder funcionar invisible)
7. Terminar el simulador



**create.** Requisito 1.

**addFossil.** Requisito 2.

**addPath.** Requisito 3.

**situate.** Requisito 4.

**mergePaths.** Requisito 5.

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

**finish.** Requisito 7 .

---

Evolution
<pre> + _(sequence : String) : Evolution + addFossil(sequence : String) : void + addPath() : void + situate(fossil : int, path : int, position : int) : void + mergePaths(first : int, second : int) : void + makeVisible() : boolean + makeInvisible() : boolean + finish() : void + ok() : boolean </pre>

## REQUISITOS DE USABILIDAD

---

- El simulador debe representar la estructura genética del mundo, la estructura genética de los fósiles por ubicar y las líneas de evolución actuales.
- Los nucleótidos se deben visualizar con figuras cuyo color identifica su tipo.
- En las líneas de evolución deben presentarse los fósiles en orden cronológico de aparición. El más antiguo debe ir de primero.
- Las líneas de evolución y los fósiles por ubicar se identifican con números de 1 a n, de izquierda a derecha y de abajo a arriba, considerando la posición en el simulador
- Si el usuario comete algún error se le debe presentar un mensaje especial, sólo si el simulador es visible.
- El simulador al iniciar es invisible

## REQUISITOS DE DISEÑO Y CONSTRUCCIÓN

---

- En su desarrollo debe respetar las decisiones de diseño presentes en este diagrama de clases
- El método **ok** retorna si la última operación se pudo realizar o no.
- Las clases se deben construir reutilizando los componentes del proyecto shapes que sean necesarios.
- Las clases deben tener la documentación estándar de java. No olvidar revisar la documentación generada.
- Las clases se deben construir en **BlueJ**. El nombre del nuevo proyecto debe ser **Evolution**

## 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.**

Publicar productos a revisión de pares	:	Miercoles 26 de agosto
Presentar revisión de pares	:	Jueves 27 de agosto
Publicar productos versión final	:	Lunes 31 de agosto (antes de las 11:00 a.m.)
Presentación final	:	Lunes 31 de agosto