

# ESCUELA COLOMBIANA DE INGENIERÍA

## PROGRAMACIÓN ORIENTADA A OBJETOS

### PROYECTO INICIAL Ciclo No. 3 2020-01

#### REFACTORING Y EXTENSIÓN

El proyecto inicial tiene como propósito desarrollar una aplicación que permita simular una situación inspirada en el **Problema C** de la maratón de programación internacional 2019 **Checks Post Facto**. En esta versión vamos a tener dos zonas: la zona de juego y la zona de configuración.

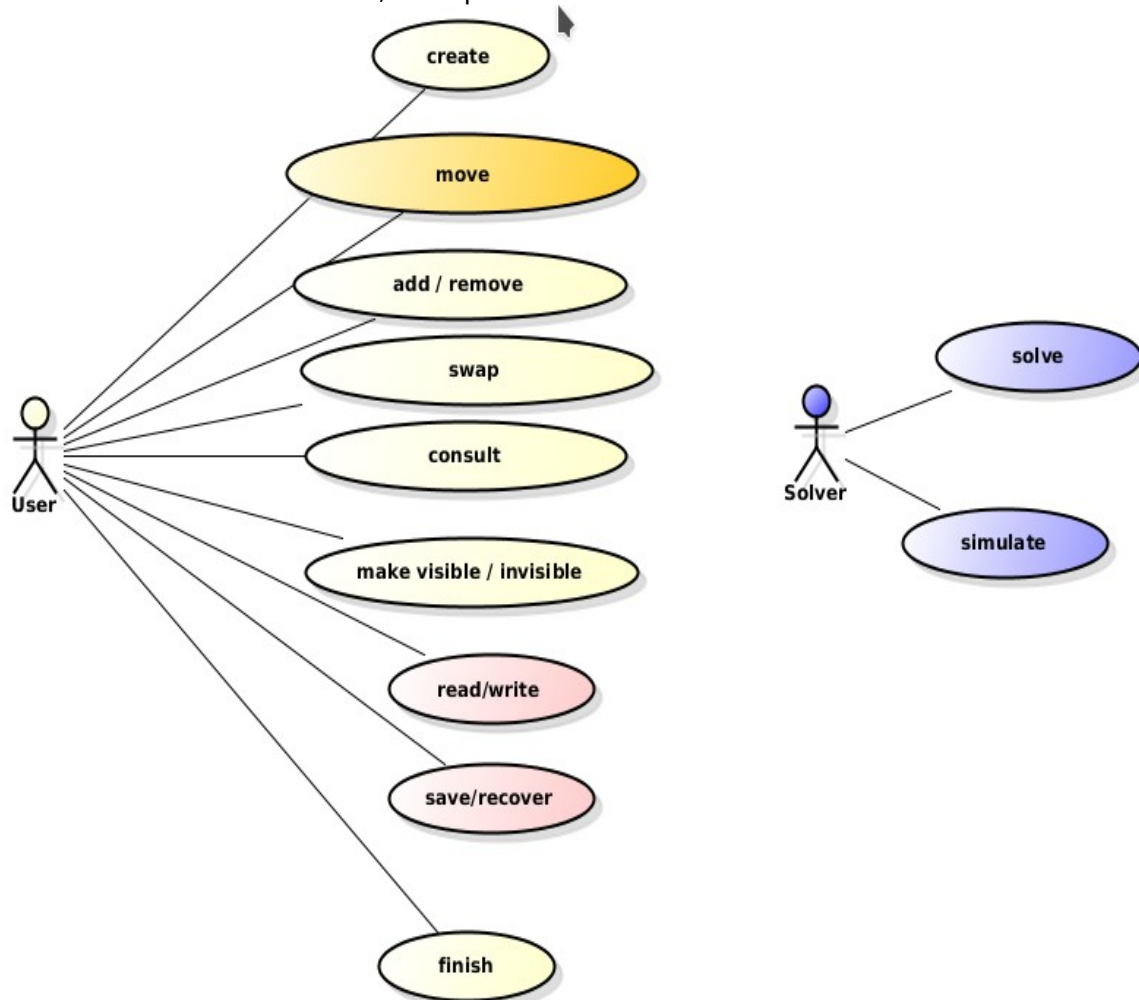
#### TERCER CICLO

El objetivo de este ciclo es extender el simulador para cumplir nuevos requisitos y perfeccionarlo para garantizar la calidad del mismo considerando los criterios de corrección y extensibilidad.

En esta entrega SI deben resolver el problema de la maratón

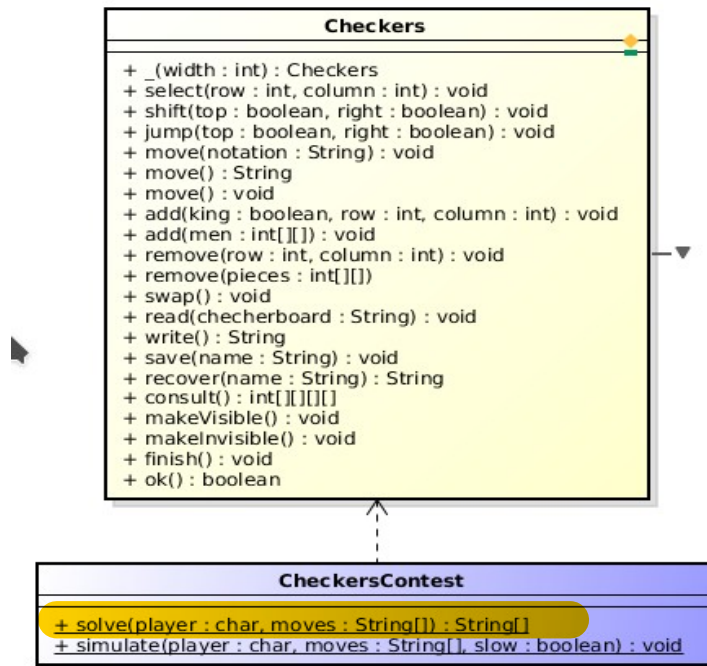
#### NUEVOS REQUISITOS FUNCIONALES

11. Debe solucionar el problema de la maratón
12. Debe simular la solución, si es posible.



**solve.** Requisito 11  
**simulate.** Requisito 12

## REQUISITOS DE DISEÑO Y CONSTRUCCIÓN



La simulación debe dejar el tablero inicial en la zona de configuración y el tablero final en la zona de juego. Si *slow* es verdadero, debe presentar en la zona de juego cada uno de los movimientos desde el tablero inicial hasta el tablero final.

**La clase Checkers se debe usar únicamente para simular NO para resolver el problema de la maratón.**

Los productos esperados en 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 completo de cada uno de los métodos de las dos clases del diseño.  
**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 programación java.  
**No olviden que el código de los métodos no debe ocupar más de una pantalla.**
3. Código de pruebas de unidad que cubran los métodos desarrollados clases. Las pruebas se deben preparar en modo invisible. La clase **CheckersContestTest** será una creación colectiva usando el wiki correspondientes  
No olviden diseñar las pruebas considerando dos preguntas: ¿qué debería hacer? ¿qué no debería hacer?  
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. (7 preguntas 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 sustentación.

## 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. El .zip debe contener el diseño (.astah), la construcción (\*.java) y **la retrospectiva (.doc)**.

Publicar productos

: Semana 08 Viernes 6 de marzo