

ESCUELA COLOMBIANA DE INGENIERÍA

PROGRAMACIÓN ORIENTADA A OBJETOS

PROYECTO INICIAL Ciclo No. 2 2020-1

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.

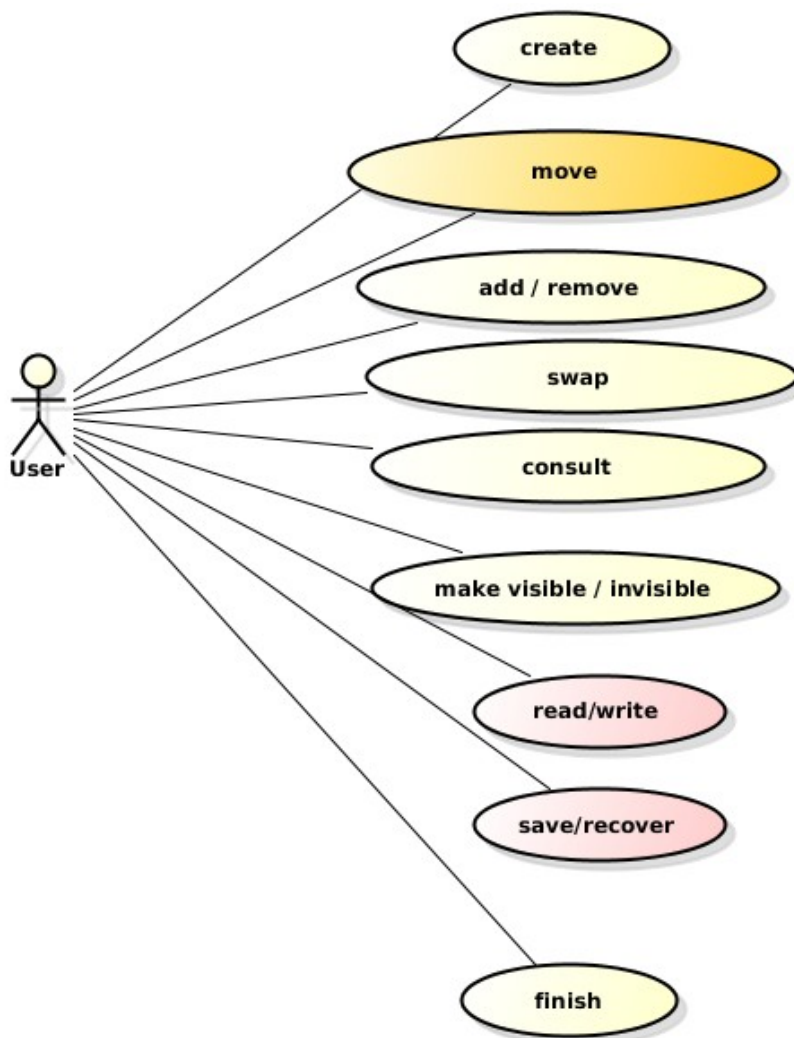
SEGUNDO CICLO

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 NO deben resolver el problema de la maratón

REQUISITOS FUNCIONALES

8. Debe permitir seleccionar y hacer un buen movimiento en el tablero de juego
9. Debe permitir leer y escribir el tablero de configuración
10. Debe permitir guardar y recuperar el tablero de configuración



- **move** Extensión.
Corresponde al requisito 9

- **read/write.** Nuevo,
Corresponde al requisito 9

- **save/recover** Nuevo,
Corresponde al requisito 11

REQUISITOS DE DISEÑO Y CONSTRUCCIÓN

Checkers
<pre>+ _(width : int) : Checkers + select(row : int, column : int) : void + shift(top : boolean, right : boolean) : void + jump(top : boolean, right : boolean) : void + move(notation : String) : void + move() : String + move() : void + add(king : boolean, row : int, column : int) : void + add(men : int[][]) : void + remove(row : int, column : int) : void + remove(pieces : int[][]) + swap() : void + read(checherboard : String) : void + write() : String + save(name : String) : void + recover(name : String) : String + consult() : int[][][] + makeVisible() : void + makeInvisible() : void + finish() : void + ok() : boolean</pre>

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

La representación en cadena para las operaciones de lectura y escritura es la definida en el enunciado de la maratón.
(Sample Output)

Los tableros se guardan y recuperan de variables de memoria identificadas por un nombre.

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 **Checkers**
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 **CheckersTestC2**.
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 **CheckersTestC2** 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.