



## SEGUNDO PARCIAL PARCIAL

12 de Octubre de 2019

### Indicaciones generales

- El examen comienza el día **12 de octubre de 2019** a las **11:00 a.m** y termina el día **14 de octubre de 2019** a las **11:59 a.m**.
- El examen es individual y de creación propia, por lo cual se espera que todos los exámenes sean diferentes.
- En e-aulas hay documentación necesaria para el desarrollo del examen, sin embargo, siéntase libre de consultar la documentación de javascript
- El examen final debe ser subido a e-aulas como un archivo llamado script.js

Modele la lógica de un tablero de ajedrez, para ello, use lo aprendido en el curso acerca de prototipos de objetos, herencia, funciones y arrays.

El modelo debe contemplar 3 prototipos, tablero, casilla y ficha. Para construirlos debe usarse funciones constructoras. A continuación se encuentran los nombres de los prototipos y sus correspondientes nombres de propiedades y métodos. El modelo propuesto será puesto a prueba en el tablero adjunto, por lo tanto, deben conservarse dichos nombres.

#### chessboard (tablero)

- **[turn]** (turno): Permite conocer cuál de los dos jugadores tiene actualmente el turno. Es de tipo string y puede tomar dos valores: blancas, negras.
- **[boxes]** (casillas): Matriz de 8x8, donde cada uno de los elementos son objetos con el prototipo casilla.

#### box (casilla)

- **[coordinate]** (coordenadas): Objeto con dos propiedades, fila y columna. Cada fila contiene números letras de la A a la H. Cada columna un número entre 1 y 8 (A1, A2, A3, ...).
- **[piece]** (ficha): Objeto de prototipo ficha, que indica la ficha que se encuentra en la casilla en ese momento. En caso de que en la casilla no haya una ficha el valor será NaN.
- **[delete]** (eliminar): Método para eliminar una ficha de la casilla, cambia el valor de la propiedad **active** de la ficha a "false".

#### piece (ficha)

- **[active]** (activo): Booleano, 0 si la ficha ha sido eliminada, 1 en otro caso.
- **[enabled]** (habilitada): booleano, 1 si la ficha tiene el mismo color del jugador actual, 0 en otro caso.



- **[box]** (casilla): Corresponde a la casilla en la que se encuentra ubicada actualmente la ficha, si ha sido eliminada su valor es NaN.
- **[color]** (color): String con la información del color de la ficha, blanca o negra. Este valor debe ser no mutable a lo largo de una misma partida.
- **[name]** (nombre): String con el nombre de la ficha pawn (peón), rook (torre), knight (caballo), bishop (alfil), king (rey) y queen (reina).
- **[movement]** (movimiento): Método que contiene la lógica del movimiento partiendo desde una determinada coordenada (parámetro). Es una función programada para cada ficha.
- **[possibilities]** (posibilidades): Array calculado con base en el método de movimiento. Contiene las posibles casillas a las que puede moverse la ficha. Si una casilla ya está ocupada por una ficha del mismo color, dicha casilla no debe ser parte del array.
- **[move]** (mover): Este método debe actualizar los datos de la partida, tiene como parámetro la casilla nueva. Cuando una ficha se mueve a la casilla nueva, la casilla actual pasa a ser NaN y la ficha.casilla pasa a tener la casilla actual. Si en la casilla nueva hay una ficha de otro color, dicha ficha debe ser eliminada.

Adicional a estos prototipos, se debe crear una función por cada una de las fichas para modelar la forma en la que se mueve cada una de las ficha (pawn (peón), rook (torre), knight (caballo), bishop (alfil), king (rey) y queen (reina)). Cada una de estas funciones que serán ubicadas dentro del método "movimiento" de la ficha una vez instanciado el objeto.