

PANDEMIA:

Pandemia es el nombre de un juego de mesa en el que varios jugadores intentan evitar que una plaga mortal se esparza por todo el planeta. En nuestra versión del juego, que será implementada en Gobstones, cada celda del tablero representa una ciudad, que puede estar infectada o no (dada por el nivel de virulencia) y el tablero representa el mundo del juego

Tres enfermedades distintas compiten para intentar ser la pandemia global que destruya a la humanidad, estas son: **Dengue, Coronavirus y Sarampión**. El juego es colaborativo, así que en cada turno los jugadores llevan adelante acciones para combatir las enfermedades. El juego se gana al erradicar todas las enfermedades, y se pierde si alguna enfermedad se transforma en una pandemia, porque en ese caso habría que entrar en cuarentena.

Se cuenta con las siguientes operaciones primitivas:

- **sarampión, coronavirus, dengue**
representan cada una de las enfermedades posibles.
El resultado es un Color que representa la enfermedad correspondiente
- **hayVirusDe_**
que dado un parámetro con la enfermedad a controlar, indica si la ciudad actual está infectada por ella.
- **virulenciaDe_**
que dado un parámetro con la enfermedad a controlar, describe el nivel de virulencia de la misma en la ciudad actual como un número.
Si la ciudad no está infectada con la enfermedad a controlar, describe 0.
- **enfermedadEnCiudadActual**
describe la enfermedad que se encuentra en la ciudad actual.
Como precondition, la ciudad debe estar infectada con alguna enfermedad.
- **lleguéALaÚltimaCiudad**
indica si el cabezal se encuentra sobre la última ciudad del mapa.
- **AumentarEnfermedad_En_**
que dado un parámetro con una enfermedad y un número, aumenta en la ciudad actual la virulencia de la enfermedad dada, tantas unidades como indique el número dado.
Como precondition, la ciudad actual no debe estar infectada con otra enfermedad diferente.
- **BajarVirulenciaDe_**
que dado un parámetro con una enfermedad, disminuye en la ciudad actual. en uno la virulencia de la enfermedad.
Como precondition, la ciudad actual debe estar infectada por la enfermedad dada.
- **IrAPrimeraCiudad**
posiciona el cabezal sobre la primera ciudad del mapa.

- **IrAPróximaCiudad**

posiciona el cabezal sobre la ciudad del mapa siguiente a la actual. Como precondition, debe existir una ciudad próxima.

1) Implementar la función **esCiudadSana**.

Propósito: Indicar cuando la ciudad actual no está infectada por ninguna enfermedad.

Precondición: Ninguna.

Resultado: ...

2) Implementar la el procedimiento **IrAPrimerCiudadCon_**.

Propósito: Posiciona el cabezal en la primer ciudad que esté infectada con la enfermedad dada

Precondición: Existe al menos una ciudad infectada con la enfermedad dada.

Parámetros: ...

3) Implementar la función **virulenciaLindanteTotalDe_**.

Propósito: Describe la cantidad total de virulencia de la enfermedad dada que acecha a la ciudad actual, teniendo en cuenta las ciudades lindantes que existan en las direcciones ortogonales.

Precondición: Ninguna.

Parámetros: ...

Resultado: ...