

# Segundo Proyecto: Concumon Go

## 75.59 - Técnicas de Programación Concurrente I

---

### Objetivo

La empresa *Miatic* está desarrollando un nuevo juego de realidad aumentada llamado *Concumon Go* y necesita, por razones de eficiencia, crear un simulador para poder testear distintos parámetros del juego. Se pide entonces implementar un simulador que cumpla con los siguientes requisitos dados a continuación.

### Requerimientos Funcionales

Los requerimientos funcionales son los siguientes:

1. El juego se desarrolla sobre una grilla que representa el mapa. Cada casillero representa una "zona" distinta.
2. Cada jugador se loggea al juego antes de empezar mediante un servidor. El servidor mantiene la cantidad máxima de jugadores loggeados.
3. Cuando un jugador intenta loggearse en el servidor, puede quedar en espera si se ha excedido la cantidad máxima de jugadores loggeados, con lo cual solamente podrá ingresar cuando uno de los jugadores existentes sale del juego. Los jugadores deben entrar al juego en el mismo orden en que intentaron loggearse, para evitar problemas de *starvation*.
4. Cuando un jugador ingresa al juego, aparece en una posición al azar dentro de la grilla o mapa.
5. Los jugadores se desplazan de una zona del mapa a otra zona aledaña cada cierto tiempo azaroso. Si al moverse se encuentra un *Concumon* el jugador lo atrapa y suma puntos. El *Concumon* atrapado se elimina del mapa.
6. Dos (o más) jugadores no pueden atrapar el mismo *Concumon*.
7. Cada posición de la grilla (o mapa) contiene a lo sumo un único *Concumon*.
8. Los *Concumons* se mueven por la grilla igual que los jugadores, a una posición aledaña, cada cierto período fijo de tiempo.
9. Existe además un nido que genera nuevos *Concumons*. Existe una cantidad máxima de *Concumons* que puede haber en la grilla en todo momento. Cuando el nido crea un *Concumon*, aparece en una posición random dentro de la grilla.
10. El *Sysadmin* verifica las estadísticas de los jugadores (puntaje) cada cierto período aleatorio de tiempo.
11. Los siguientes parámetros deben ser configurables:
  - a) Las dimensiones de la grilla
  - b) El intervalo de tiempo en el cual se mueven los *Concumons*
  - c) La cantidad máxima de jugadores loggeados en el juego
  - d) La cantidad máxima de *Concumons* presentes en la grilla en todo momento

## Requerimientos no Funcionales

Los siguientes son los requerimientos no funcionales de la aplicación:

1. El proyecto deberá ser desarrollado en lenguaje Haskell utilizando el modelo de concurrencia provisto por dicho lenguaje.
2. La simulación puede no tener interfaz gráfica y ejecutarse en una o varias consolas de línea de comandos.
3. La aplicación deberá funcionar en una única computadora.

## Tareas a Realizar

A continuación se listan las tareas a realizar para completar el desarrollo del proyecto:

1. Dividir el proyecto en threads. El objetivo es lograr que la simulación esté conformada por un conjunto de threads que sean lo más sencillos posible.
2. Diseñar las estructuras de datos necesarias para que los threads definidos puedan comunicarse entre sí.
3. Diseñar el skeleton code necesario para que funcionen los threads y las estructuras de datos definidos anteriormente. Como tal, el skeleton code no debe implementar la funcionalidad requerida sino simplemente definir las bases necesarias para que funcione la aplicación.
4. Realizar la codificación de la aplicación. El código fuente debe estar documentado.

## Entrega

La entrega del proyecto comprende lo siguiente:

1. Informe, se deberá presentar impreso en una carpeta o folio y en forma digital (PDF) a través del campus
2. El código fuente de la aplicación, que se entregará únicamente mediante el campus

La entrega en el campus estará habilitada hasta las 19 hs de la fecha indicada oportunamente.

El informe a entregar debe contener los siguientes items:

1. Documentación de los threads y las estructuras de datos definidas.
2. Detalle de resolución de la lista de tareas anterior.
3. Diagrama que refleje los threads y estructuras de datos definidos, el flujo de comunicación entre ellos y los datos que intercambian.