

---

## Práctica 3: Hormiga de Langton Generalizada

---

### Objetivo

El objetivo de esta práctica es comprender el funcionamiento de la herencia y el polimorfismo en Orientación a Objetos, y utilizarlos adecuadamente en un programa en lenguaje C++.

### Entrega

Esta práctica se realizará en dos sesiones de laboratorio en las siguientes fechas:

Sesión tutorada: 6, 7 y 8 de marzo de 2018;

Sesión de entrega: 13, 14 y 15 de marzo de 2018.

### Enunciado

En esta segunda práctica sobre la Hormiga de Langton [1] se quieren trabajar los mecanismos de herencia y polimorfismo en el lenguaje C++. Para ello se generaliza el concepto de **Hormiga**, visto en la práctica anterior [3], convirtiéndolo en un concepto abstracto que permita particularizar en distintos **Tipos de Hormigas**, cada uno de los cuales presenta un comportamiento diferente. Se pide implementar una aplicación en la que coexistan, de forma simultánea, sobre la misma Rejilla, los distintos Tipos de Hormigas.

Turk y Propp [4] propusieron una generalización de las reglas clásicas para permitir que los cuadrados de la Rejilla tomen más de dos colores. Cada color se asocia a un sentido de giro ( $90^\circ$  a la derecha 'D' ó  $90^\circ$  a la izquierda 'I'). Cada Tipo de Hormiga se identifica con una cadena de D's ó I's. La  $i$ -ésima posición de la cadena indica el sentido de giro de ese tipo de hormiga cuando se encuentra sobre un cuadrado de color  $i$  ( $0 < i < n$ ), siendo  $n$  el número de colores.

Así, por ejemplo, el Tipo de Hormiga "DDII" se mueve sobre una rejilla bidimensional cuyas celdas se alternan entre 4 colores. En los dos primeros colores la hormiga gira a la derecha, y en los dos últimos colores gira a la izquierda. Siguiendo esta nomenclatura a la versión clásica de la hormiga con dos colores le corresponde el tipo "DI".

Esta nomenclatura no recoge los cambios de color que provoca una hormiga, por lo que esta decisión se deja al programador.

### Notas de implementación

Los objetos utilizados en la primera práctica requieren los siguientes cambios:

- Objeto **Rejilla**, contiene los cuadrados de 4 colores sobre los que se mueve la hormiga. Los colores se representan mediante un tipo enumerado donde se haga corresponder el color blanco con el valor 0. Inicialmente se definirá una Rejilla con todos los cuadrados de color blanco. De forma opcional, se podrían definir otras situaciones iniciales sobre la Rejilla con algunos cuadrados de otros colores. Para evitar trabajar con un entorno gráfico, la visualización de un cuadrado blanco se hará con el carácter ' ', y para la visualización de un cuadrado de otro color se le asignará otro carácter a cada color. El objeto Rejilla

oculta los detalles de su implementación, esto es, la estructura de datos utilizada para guardar los cuadrados de colores, y provee las operaciones necesarias para consultar y cambiar el color de cualquiera de los cuadrados que contiene.

- Clase abstracta **Hormiga**, es la clase base de la cual derivan todos los tipos de hormigas. En esta clase se definen, mediante métodos nulos, los comportamientos genéricos que cada tipo de hormiga debe particularizar. Toda Hormiga es responsable de guardar su dirección de movimiento: Izquierda, Derecha, Arriba o Abajo; y la posición del cuadrado sobre el que se encuentra ubicada. Toda Hormiga es responsable de implementar su movimiento, para lo cual recibe el color del cuadrado que está ocupando sobre la Rejilla y retorna el color al que debe cambiar dicho cuadrado. Toda Hormiga es responsable de visualizarse sobre la Rejilla.

```
class Hormiga {  
    ...  
    virtual Color movimiento(Color) = 0;  
    virtual ostream& ver(ostream&) const = 0;  
    ...  
};
```

- Objeto **Hormiga Tipo "X"** definido mediante la clase **HormigaX** que derivada de la clase abstracta **Hormiga**. Implementa las reglas de movimiento definidas para la Hormiga Tipo "X", y determina la forma de visualizar a este tipo de hormigas sobre la Rejilla. Para evitar trabajar con un entorno gráfico, la visualización de una hormiga utilizará un carácter que identifique de forma única su tipo y dirección de movimiento.
- El objeto **Máquina** es el responsable de contar y gestionar los pasos de la simulación, así como de mantener el contador del número de hormigas instanciadas de cada tipo. En cada paso cada una de las hormigas realiza un movimiento y se actualiza la visualización de la Rejilla y las hormigas. En pantalla se visualizará un cuadrado de dimensiones MxN donde se encuentren las hormigas, o parte de ellas. La Máquina establecerá la situación inicial de la Rejilla y de las hormigas instanciadas. Luego comenzará a realizar los pasos de la simulación hasta que se alcance la situación de parada, que puede ser por un número máximo de pasos o cuando el usuario detenga la simulación.

Durante las sesiones de laboratorio se podrán proponer modificaciones y mejoras en el enunciado de la práctica.

## Referencias

- [1] Wikipedia. Hormiga de Langton: [https://es.wikipedia.org/wiki/Hormiga\\_de\\_Langton](https://es.wikipedia.org/wiki/Hormiga_de_Langton)
- [2] Wikipedia. Christopher Langton: [https://es.wikipedia.org/wiki/Christopher\\_Langton](https://es.wikipedia.org/wiki/Christopher_Langton)
- [3] Práctica 2. AEDA curso 2017-2018: [https://campusvirtual.ull.es/1718/pluginfile.php/223640/mod\\_resource/content/12/AEDA-pract02.pdf](https://campusvirtual.ull.es/1718/pluginfile.php/223640/mod_resource/content/12/AEDA-pract02.pdf)
- [4] Gale, D.; Propp, J.; Sutherland, S.; Troubetzkoy, S. (1995). «Further Travels with my Ant». Mathematical Intelligencer 17: 48-56.