

EIF204 – Programación 2

Proyecto de programación #2

OBJETIVO DEL PROYECTO

Crear una simulación de un ecosistema con diferentes tipos de criaturas (herbívoros, carnívoros, omnívoros), recursos (agua, plantas, etc.) y un entorno cambiante (clima, estaciones). Se debe diseñar un sistema extensible y mantenible aplicando diversos patrones de diseño.

DESCRIPCIÓN DEL PROYECTO

El programa por desarrollar simulará un ambiente limitado que incluye diferentes elementos que interactúan entre sí. El sistema definirá una cuadrícula de espacio virtual que permita colocar los diferentes elementos en distintas posiciones del sistema y que estos se desplacen por el ambiente. Las posiciones pueden ser indicadas usando coordenadas enteras en dos dimensiones (posición horizontal y vertical).

Los elementos o componentes por definir dentro de la simulación incluirán:

- **Criaturas**
 - Deben tener comportamientos diferentes según su tipo (herbívoros, carnívoros, omnívoros).
 - Deben poder moverse (desplazarse en el sistema), buscar recursos, alimentarse (adquirir recursos) y reproducirse (duplicarse según su edad en el sistema y nivel de energía). Las criaturas consumen recursos al desplazarse y reproducirse. Cuando una criatura agota sus recursos, “muere” y es eliminada del sistema. Cuando una criatura “nace” en el sistema, tiene un nivel inicial de energía que se va reduciendo hasta que la criatura consume un recurso o muere.
- **Recursos**
 - Se regeneran en el entorno con ciertas reglas (plantas crecen con sol y agua, por ejemplo). Al morir una criatura, se genera una unidad de recurso consumible, y pueden generarse unidades en puntos específicos del sistema a intervalos regulares. Al ser consumidos, los recursos aumentan el nivel de energía de las criaturas.
- **Entorno**
 - Simulación por *ticks* de tiempo (intervalos discretos que pueden representar cualquier unidad de tiempo: segundos, minutos, etc.). El sistema debe ejecutarse por una cantidad máxima de *ticks* o se permitirá que el usuario indique su finalización de manera interactiva.
 - Las condiciones del entorno afectan a los recursos y criaturas. El entorno tiene dimensiones limitadas y un nivel de energía total que puede fluctuar. El nivel de energía determina los recursos disponibles.

- El programa debe poder resguardar y recuperar el estado del sistema en cualquier momento.
- **Interfaz de usuario**
 - La interfaz del programa utilizará comandos de consola que permitan ver el estado del en cada ciclo y modificar sus parámetros.

Para cada tipo de elemento (criatura o recurso) deben definirse al menos 3 elementos con diferentes características, y cada elemento ocupará una posición en el sistema. Sólo podrá existir un elemento en una posición determinada del sistema. Es decir, una criatura no puede moverse a una posición donde ya exista otra criatura y si el espacio contiene un recurso, este debe ser consumido para poder ocupar la posición correspondiente.

CONSIDERACIONES DE IMPLEMENTACIÓN

Escriba el programa utilizando el lenguaje de programación C++, utilizando una interfaz de consola.

Debe considerar el uso de patrones en la solución:

- ***Factory Method (Abstract Factory)***

Se utiliza una fábrica (*factory*) para crear diferentes tipos de criaturas.
- ***Strategy***

Para definir los comportamientos de movimiento, alimentación y reproducción.
- ***Observer***

Para que las criaturas y recursos reaccionan a cambios en el entorno.
- ***Singleton***

Como controlador del entorno que mantiene el estado global del sistema.
- ***Composite***

Sirve para modelar la jerarquía de objetos del ecosistema (grupo de criaturas, recursos agrupados).
- ***Iterator***

Para modificar el estado de los conjuntos o colecciones de elementos.
- ***Decorator (opcional)***

Permite añadir habilidades a las criaturas (velocidad, modos de reproducción) en tiempo de ejecución.

Toda la información de la aplicación debe resguardarse en archivos para poder utilizarla entre distintas ejecuciones del programa. El formato utilizado para los datos queda a criterio de los estudiantes, pero sí se utilizarán archivos de texto, NO binarios.

Incluya en el código un bloque de información sobre el proyecto y las personas que presentan la solución, **en cada uno de los archivos fuente**, de la siguiente manera:

```
/* -----+
 *
 * (c) 2025
 * EIF204 - Programación 2
 * 1er ciclo 2025
 * NRC 9999 - Grupo 00
 * Proyecto 2
 *
 * 1-2345-6789; Hernández Zúñiga, Adalberto
 * 9-8765-4321; Méndez Rojas, Adelaida
 * (cédula y nombre de la persona)
 * (cédula y nombre de la persona)
 *
 * versión 1.0.0 2025-06-01
 *
 * -----+
 */
```

Agregue además los comentarios necesarios para explicar la implementación de cada función definida, e incluya las pruebas que considere necesarias para verificar su correcto funcionamiento.

ENTREGA Y EVALUACIÓN

El proyecto debe entregarse **por medio del aula virtual, en el espacio asignado para ello**. La entrega se hará al finalizar la **semana de evaluaciones finales. (viernes 13 de junio de 2025)**. No se aceptará ningún proyecto después de esa fecha, ni se admitirá la entrega del proyecto por correo electrónico. El proyecto se puede realizar en grupos de **cuatro personas, como máximo**.

Deberá entregar:

- Diagramas de clases UML correspondientes al código.
- Código en C++ con procedimientos adecuados de prueba.
- Informe explicando la aplicación de los diferentes patrones de diseño utilizados (no es obligatorio todos los patrones, pero debe justificar cuáles utiliza o no según las necesidades del programa).
- Detalle del proceso de prueba de la aplicación.

En caso de que las funciones no trabajen adecuadamente, efectúe un análisis de los resultados obtenidos, indicando las razones por las cuales el programa no se ejecuta correctamente, y cuáles son las posibles correcciones que se podrían hacer.

El proyecto se evaluará de acuerdo con la siguiente ponderación:

Rubro	Puntos
Uso correcto de los patrones aplicados	30
Diseño y estructura de la aplicación	20
Funcionalidad	20
Organización del código	15
Persistencia	15
Funcionalidad opcional	10
Total	100+10

Observaciones generales:

- Se debe incluir en **cada documento entregado** el nombre completo y cédula de cada participante del grupo, indicando el nombre del curso, ciclo lectivo y descripción del trabajo que se entrega, de la manera indicada.
- Cualquier trabajo práctico que no sea de elaboración original de los estudiantes (plagio) se calificará con nota 0 (cero) y se procederá como lo indiquen los reglamentos vigentes de la universidad.
- Los trabajos que se reciban después de la fecha señalada para su entrega, **en caso de ser aceptados, serán penalizados con un 30% de la nota por cada día de atraso.**