

# Proyecto #3 - AntVillage

Instituto Tecnológico de Costa Rica  
Área de Ingeniería en Computadores  
CE1103 - Algoritmos y Estructuras de Datos I  
Primer Semestre 2022  
Valor 15%



## Objetivo General

- Implementar una solución a un problema mediante la utilización de estructuras de datos lineales y grafos.

## Objetivos Específicos

- Diseñar soluciones utilizando el paradigma orientado a objetos.
- Utilizar UML para modelar la solución de un problema.
- Aplicar **patrones de diseño** en la solución de un problema.
- Implementar estructuras de datos lineales y grafos.
- Desarrollar las capacidades relacionadas con el trabajo individual y en equipo y el diseño de soluciones.

## Descripción del problema

El proyecto consiste en la implementación de un juego ambientado en una aldea de hormigas.

### *Inicio del juego*

Al inicio del juego se debe seleccionar la cantidad de nodos que tendrá la aldea y a partir de este dato se debe generar el grafo dirigido con pesos, generando aristas y pesos de forma aleatoria.

Antes de iniciar el juego, también se debe definir la cantidad de unidades de alimento que debe conseguir cada comunidad de hormigas, para ganar el juego.

### *Reglas del juego*

Para la implementación de este juego debe considerar los siguientes lineamientos:

- Las hormigas pueden desplazarse a través de los nodos, para tratar de conseguir alimento.
- Existen dos grupos de hormigas, las hormigas azules y las hormigas verdes.
- Cada grupo tiene una hormiga líder que es la encargada de recorrer la aldea.
- Las hormigas verdes recorren la aldea usando el algoritmo Dijkstra. Se debe mostrar el desplazamiento, en tiempo real, que realiza la hormiga para intentar llegar a la comida. Las hormigas verdes **saben dónde está la comida** cada vez que aparece en la aldea.
- Las hormigas azules recorren la aldea usando un algoritmo de fuerza bruta. Se debe mostrar el desplazamiento, en tiempo real, que realiza la hormiga para intentar llegar a la comida. Las hormigas azules **no saben dónde** está la comida cada vez que aparece en la aldea.

- Este juego lo que permite es mostrar la simulación del recorrido de las hormigas para intentar conseguir la mayor cantidad de alimento.
- Se debe mostrar, en tiempo real, la cantidad de alimento que ha recolectado cada hormiga.
- En el nodo donde se encuentra la comida que deben buscar las hormigas, se debe mostrar una imagen relacionada al alimento que están buscando.
- Cada vez que una hormiga logra conseguir un alimento (llega al nodo de primero), debe aparecer otro alimento en la aldea. La posición donde debe aparecer el nuevo alimento es definida por el usuario al seleccionar un nodo. Esto puede realizarlo antes de que una hormiga consiga el alimento, pero solo permite seleccionar la posición del siguiente alimento.

#### *Fin del juego*

- El juego finaliza cuando una hormiga alcanza la cantidad máxima de alimento que se había establecido antes de iniciar el juego.
- Cada vez que termina el juego, se debe crear un archivo XML, donde se almacene la cantidad de alimento recolectada por cada comunidad de hormigas. Es decir, existe un archivo XML por cada partida del juego.

#### *Algoritmo de fuerza bruta*

Para este algoritmo, lo que debe hacerse es ir recorriendo el grafo, bajo algún criterio definido por cada grupo. Cuando se llegue a un nodo donde no se pueda agregar un nodo diferente a la ruta (ruta simple), se debe devolver en la ruta que estaba usando y buscar un nodo donde pueda tomar una decisión diferente a la que ya había tomado.

#### **Ptos Extra (5%)**

Implementar la funcionalidad de visualizar el histórico de partidas, donde permita recorrer las partidas (de las más recientes a las más antiguas, y ver la información de la cantidad de alimento recolectada por cada comunidad de hormiga. Con una opción de siguiente o atrás, puedo recorrer este histórico de partidas. Para esto debe implementarse una lista doblemente enlazada circular, donde cada nodo representa una partida. Cada archivo XML debe cargarse en un nodo de esta lista doblemente enlazada circular. Adicionalmente, a través de la interfaz gráfica, se pueden eliminar los registros de partidas, con lo cual se elimina de la lista y se borra el archivo XML asociado a esa partida.

#### **Documentación requerida**

1. Internamente, el código se debe documentar siguiendo los estándares de documentación para cada lenguaje utilizado.
2. La documentación externa se hará en un documento que incluya lo siguiente (deberá entregarse un PDF):
  - a. Breve descripción del problema.
  - b. Diagrama de clases.
  - c. Descripción de las estructuras de datos desarrolladas.
  - d. Protocolo de comunicación entre cliente y servidor
  - e. Problemas encontrados en forma de bugs de *github*: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.

3. **Planificación y administración del proyecto:** se utilizará Jira para la administración del proyecto. Debe incluir:
  - a. Lista de features e historias de usuario identificados de la especificación.
  - b. Plan de iteraciones que agrupen cada bloque de historias de usuario de forma que se vea un desarrollo incremental
  - c. Descomposición de cada user story en tareas.
  - d. Asignación de tareas entre los integrantes del grupo.

**Aspectos operativos y evaluación:**

1. **Fecha de entrega: De acuerdo con el cronograma del curso y lo establecido en el TEC Digital**
2. El proyecto tiene un valor de 15% de la nota del curso.
3. El trabajo es **en grupos de máximo 4 personas.**
4. Deben entregar en el TEC Digital un documento con el link del repositorio de GitHub, Jira, el PDF de la documentación y el PDF del Anexo. En ambas herramientas deben dar acceso al correo del profesor.
5. Es obligatorio utilizar un Git y GitHub para el control de versiones del código fuente y evidenciar el uso de Commits frecuentes.
6. Es obligatorio integrar toda la solución.
7. El código tendrá un valor total de 70%, la documentación externa 20% y la defensa un 10%.
8. De las notas mencionadas en el punto anterior se calculará la Nota Final del Proyecto.
9. Se evaluará que la documentación sea coherente, acorde a la dificultad/tamaño del proyecto y el trabajo realizado. Se recomienda que realicen la documentación conforme se implementa el código.
10. La nota de la documentación externa es proporcional a la completitud del proyecto.
11. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
12. Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la primera cita de revisión oficial.
13. Aun cuando el código y la documentación externa tienen sus notas por separado, se aplican las siguientes restricciones
  - a. Si no se entrega documentación externa, automáticamente se obtiene una nota de cero en la nota final del proyecto.
  - b. **Si no se entrega el Anexo se obtiene una nota de cero en la nota final del proyecto. El Anexo debe entregarse completo. Si tienen alguna duda con respecto a las partes del documento deben aclararlo con el profesor.**
  - c. Si no se utiliza un manejador de código se obtiene una nota de cero en la nota final del proyecto.
  - d. Si la documentación externa no se entrega en la fecha indicada se obtiene una nota de cero en la nota final del proyecto.
  - e. Si el código no compila se obtendrá una nota de cero en la nota final del proyecto, por lo cual se recomienda realizar la defensa con un código funcional.
  - f. El código debe ser desarrollado Java (Windows), caso contrario se obtendrá una nota de cero en la nota final del proyecto.

14. La revisión de la documentación será realizada por parte del profesor. Podría ser revisada, por parte del profesor, antes o después de la cita de revisión del proyecto. Durante la defensa del proyecto podría revisarse el diagrama de clases, la documentación interna y la documentación en el manejador de código.
15. Cada estudiante tendrá como máximo 15 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
16. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
17. Cada estudiante es responsable de llevar los equipos requeridos para la revisión, si no cuentan con estos deberán avisar al menos 2 días antes de la revisión al profesor para coordinar el préstamo de estos.
18. Durante la revisión únicamente podrán participar el estudiante, asistentes, otros profesores y el coordinador del área.

## **ANEXO DEL PROYECTO**

### **Objetivo General**

- Elaborar un documento que evidencie la participación activa de los integrantes del equipo de acuerdo con el rol asignado y la ejecución adecuada de los lineamientos establecidos por el grupo para el trabajo en equipo.

### **Objetivos Específicos**

- Participar activamente en el equipo de trabajo de acuerdo con el rol asignado.
- Ejecutar adecuadamente los lineamientos para el trabajo en equipo (metas, roles, reglas, cronogramas, bitácoras, otros).

### **Atributos de Acreditación**

- Trabajo individual y en equipo (Medio).

### **Descripción del Entregable**

Cada grupo debe elaborar un documento que tenga la siguiente estructura:

1. Portada.
2. Tabla de contenidos.
3. Introducción.
4. Trabajo individual y en equipo (evidencia de la participación activa en el equipo de trabajo de acuerdo con el rol asignado y la ejecución adecuada de los lineamientos para el trabajo en equipo (metas, roles, reglas, cronogramas, bitácoras, otros):
  - a. Metas del proyecto: Describir los entregables asociados al proyecto.
  - b. Roles: Describir los roles y responsabilidades utilizados en el desarrollo del proyecto.
  - c. Reglas: Describir las reglas principales que han sido definidas para el trabajo en equipo.

- d. Cronograma: Plan de proyecto con las actividades planeadas, responsables de cada actividad y fechas de entregas estimadas.
- e. Minutas de las sesiones de trabajo: De manera que se evidencie la participación activa en el equipo de trabajo de acuerdo con el rol asignado y el seguimiento al plan de trabajo.
- f. Bitácora: Evidencia la participación activa en el trabajo en equipo y la ejecución adecuada de los lineamientos para el trabajo en equipo, de cada uno de los integrantes del equipo. Deben describir las actividades realizadas como reuniones con el compañero de trabajo, investigaciones, consultas, entre otras. Se debe describir todo por más insignificante que sea. Esto demostrará el trabajo de cada uno de los miembros del equipo según el rol asignado.

#### **Aspectos operativos y evaluación**

1. El documento debe ser enviado en formato PDF en conjunto con los entregables del proyecto.