

# Un proto-calabozo

## Curso Estructuras de datos Especificación de Proyecto

por Aurelio Sanabria

### Motivación

Tal como vimos en clase, no siempre, la etapa de abstracción y modelado de los datos para aprovechar las oportunidades y ventajas que nos provee una estructura de datos específica es super clara y directa. Como ejemplo de esto en este proyecto vamos a construir un juego de calabozo (meh...más o menos). De acuerdo a wikipedia:

*«Los videojuegos de mazmorras (también llamados videojuegos de calabozos o roguelike en América latina) son un subgénero de los videojuegos de rol que se caracterizan por una aventura a través de laberintos, a través de niveles generados por procedimientos al azar, videojuegos basados en turnos, gráficos basados en fichas y la muerte permanente del personaje del jugador. La mayoría de los videojuegos de mazmorras se basan en una narrativa de fantasía alta, que refleja su influencia de los juegos de rol de mesa como Dungeons & Dragons.»*

Tons demosle

### Objetivos Formativos

El presente proyecto tiene correspondencia con los siguientes objetivos específicos establecidos en el programa del curso de *Estructuras de datos* (presentado al inicio del semestre):

- Implementar soluciones utilizando las diversas estructuras de datos aprendidas
- Elaborar diseños detallados de soluciones basados en estructuras de datos que le permitan resolver problemas de la vida real
- Determinar buenas prácticas de programación para la implementación de estructuras de datos
- Explicar distintas estructuras de datos (lineales, jerárquicas, generales) con el objetivo de la resolución de problemas de mediana complejidad en la implementación y diseño.
- Elaborar estructuras de datos alternas que permitan la resolución de problemas con mayor eficacia, haciendo híbridos de las mismas con el objetivo de crear soluciones más robustas

## **Especificación del proyecto**

Tal cómo les prometí este es el proyecto más pequeño, por lo que la mayor parte de la progra será opcional.

### **A. Generar un grafo fuertemente conexo**

Se le solicita el tamaño del laberinto a la usuaria y se crea un grafo no dirigido fuertemente conexo y valorado. Donde el valor de cada arista se define de forma aleatoria.

Debe proveer al menos dos formas de imprimir el grafo.

### **B. Aplicar un algoritmo para calcular el árbol de expansión mínima**

Se aplica un algoritmo para calcular el árbol de expansión mínima (Prim o Kruskall) y con esto se generará el laberinto.

Debe proveer al menos dos formas de imprimir el grafo.

### **C. Aplicar el algoritmo de Dijkstra para determinar un camino más corto.**

Dado un nodo de inicio se calcula el algoritmo de Dijkstra y se imprime el camino más corto a cada uno de los nodos del grafo.

### **(Opcionales)**

La parte principal es solo implementar los algoritmos, pero si así lo desean se podrá implementar la funcionalidad para completar el juego de calabozo donde cada nodo del grafo representa una habitación y cada arista representa una conexión a otra habitación.

Acá existe un jugador que tiene una vida 100 y un nivel de ataque (aleatorio) asignados al inicio del juego, El jugador se coloca aleatoriamente dentro del calabozo y gana si llega a un nodo de salida en los extremos del grafo (aka sale del calabozo).

### **D. Navegar por el laberintos**

En cada habitación la jugadora puede escoger en que dirección moverse (norte, sur, este u oeste). Acá es importante destacar que debe darse cuenta cuando llegó a un nodo salida.

### **E. Peleas con Ogros**

En cada habitación puede encontrarse o no con un ogro. Los ogros se colocan en el laberinto de manera aleatoria al inicio de juego justo antes de que el usuario entre. Cada ogro tiene una cantidad de vida (aleatoria) y un poder de ataque (aleatorio). Cuando la usuaria entra a la habitación, la peleá se da automáticamente, y se determina si mató al ogro o si murió.

Ojo que se pueden implementar niveles de dificultad como: Fácil, Medio y Difícil incrementando la vida y ataque de los ogros.

### **F. Encontrarse pociones**

Al inicio del juego se colocan de forma aleatoria pociones que elevan la vida del jugador o su nivel de pelea. Es perfectamente posible que en una habitación coexistan un ogro y una poción.

### **H. Determinar la ganadora.**

Para crear la ganadora hay que ver por donde se sale del grafo, eso quiere decir que hay que revisar los nodos que tienen solo una conexión y designarla como una habitación sin final o una puerta de salida.

Si llega a una salida, ganó.

## ***Metodología***

Se espera que lxs estudiantes tomen un tiempo para que, de forma creativa, se plantee como podría solucionar el problema utilizando para esto papel y lápiz, para dibujos, esquemas y diagramas que no debe entregar pero que es recomendable hacer.

Implementar las estructuras de datos de forma independiente y verificando que funcionen es altamente recomendable antes de integrarlas al resto del proyecto, adicionalmente se recomiendan pruebas extensivas de todos los componentes del proyecto. Por último es importante aclarar que las *tablas de dispersión* y las *listas* formarán parte del segundo proyecto programado por lo que deben hacerlas de forma que sean re-utilizables.

Es importante aclarar que la dificultad de este proyecto no reside solamente en la creación de

estructuras sino también en cómo organizan la información dentro de estas de forma que:

1. Se pueda trabajar sobre estas con cierta facilidad.
2. Almacene todos los datos necesarios.
3. Permita convertir las estructuras de datos a una representación para la persona usuaria.

Se pueden discutir ideas, pros y contras, algoritmos, estrategias, recursos de internet u otros con lxs compañerxs pero NO SE PERMITE EL PLAGIO, si se detecta el plagio se asigna nota de 0 y se pierde la buena voluntad del votán.

## Rúbrica

El proyecto se calificará considerando los rubros expuestos en la tabla 1, cada rubro recibirá una puntuación entre 0 y 5 de acuerdo a lo designado en la Tabla 2. La siguiente tabla detalla algunos elementos esperados y que deben ser considerados para cada rubro.

Tabla 1: Rubros de calificación

Rubro	Descripción	Puntos
<i>Generación del grafo</i>	<i>Genera el grafo al tamaño solicitado por el usuario y permite imprimirlo de, al menos, dos formas distintas</i>	10
<i>Algoritmo de Dijkstra</i>	Permite el cálculo del algoritmo de Dijkstra y funciona bien, imprime los caminos de forma correcta.	25
<i>Algoritmo para el cálculo del árbol de expansión mínima</i>	Permite el calculo del algoritmo del árbol de expansión mínima.	25
<i>Sistema de menús</i>	Hay interfaz de línea de comandos, está validada, es funcional y se entiende.	10
<i>Puntos extra</i>	<i>Hasta un máximo de 25 puntos extra a discreción de su votán.</i>	
<i>Discusión de resultados</i>	Aportar a la discusión, tiene una visión crítica del proyecto y conocer de todas las partes del mismo. Puede explicar COMPLETAMENTE todas las partes de su código y además cumplió con todos los requisitos administrativos a cabalidad (formato del correo, git, apuntarse correctamente para la revisión, entre otros.	20
<i>Calidad de código</i>	Todas las funciones tienen comentarios de función y responden al formato de doxygen para C, hay una modularización sustancial del proyecto, no hace include de archivos .c, programa todo en español y utilizando snake_case.	10

	Además los archivos .h están documentados así cómo los structs.	
		100

*Tabla 2: Distribución de puntos por calificación*

<b>Excelente</b>	<b>Bueno</b>	<b>Regular</b>	<b>Malo</b>	<b>Deficiente</b>	<b>No existe</b>
5 puntos	4 puntos	3 puntos	2 puntos	1 punto	0 puntos

## Grupo de trabajo

- Este proyecto será en grupos de 2 personas

## Aspectos Generales

1. Fecha de entrega: 23 de Noviembre 2022
    - 10:00 p.m GMT-6
    -
- Al correo aurelio.itcr@gmail.com deben enviar un link a repositorio privado en <https://gitlab.com> donde tienen el código de la progra y un zip de un clone de su repositorio.
  - Un solo repositorio por equipo de trabajo.
  - Ambas personas estudiantes están agregadas al repositorio con permisos para hacer commits.
  - El repositorio contiene al menos 10 commits de cada integrante del grupo.
  - Se enviarán al correo aurelio.itcr@gmail.com con el asunto el el formato de clase → [2022 IIS][Datos][Proyecto 3] Nombre1 Apellido1 – Nombre2 Apellido2
  - Los trabajos de entrega tardía no se calificarán.
  - Se aclararán dudas sobre la progra vía Telegram.
  - Pueden debatir con otros grupos sobre posibles formas de resolver los problemas pero las implementaciones deben ser realizadas de forma independiente entre grupos pues deben estar preparados para defender su trabajo durante una revisión
  - Cualquier plagio implica nota de 0
  - Funciones copiadas de internet implican plagio (no importa si las adaptaron o no) y nota 0.
  - Se asume que si no puede explicar su propio código es por plagio y nota 0.