

PROYECTO SEMESTRAL
CURSO DE MÉTODOS DE
PROGRAMACIÓN

1-2021

CONTENIDO

Introducción	1
Contexto	2
Características de la simulación	3
Trabajo a realizar	6
Características de entrada y salida	6
Funcionalidades	8
Entregas y evaluación	10
5.1 Entrega 1	11
5.2 Entrega 2	12
5.3 Entrega 3	12
5.4 Entrega 4	14
5.5 Entregas en general	14

1. INTRODUCCIÓN

En este documento se presenta el enunciado del proyecto del laboratorio a construir por los estudiantes del curso de Métodos de Programación para el primer semestre del año 2021.

El proyecto está basado en la simulación de un juego, el cual debe ser programado en lenguaje C, contemplando que la interacción con el usuario debe ser mediante los dispositivos de entrada y salida estándar del computador y archivos de texto plano. También se aceptan otro tipo de visualización más sofisticada, si es que así se desea por el equipo desarrollador..

Cabe destacar que este es un juego que busca trabajar las distintas técnicas de resolución de problemas que se proponen en el curso, y que no es la forma de hacer un juego en sí, el contexto es completamente inventado, y si hay semejanzas con la realidad, es mera coincidencia. Se ha evitado el uso de nombres de personas, para que estas no puedan ser identificadas.

Por parte del cuerpo académico se espera que usted como estudiante del curso pueda plantear una solución utilizando tanto los conocimientos entregados en la cátedra, como laboratorios y los ejercicios del curso, en conjunto a su ingenio e imaginación, con el fin de conseguir un producto de una excelente calidad.

En el documento se presenta inicialmente el contexto al problema, el cual plantea una situación y muestra un problema a resolver. Posteriormente se da paso a la explicación de cómo ha de funcionar cada uno de los elementos que se ven involucrados en la simulación a implementar, contando el cómo funciona cada una de las cosas que tienen relación a la implementación solicitada y, además, dando en algunos casos elementos básicos que debe tener la solución. Esto es mostrado en la sección llamada Características de la simulación.

Luego existe una sección en donde se nombran las funcionalidades que debe tener su programa, las cuales deben ser cumplidas para la evaluación final del proyecto. El documento finaliza con la forma de evaluación y las reglas que se les solicitan en cada una de las entregas, además de las fechas de cada una y en qué consisten.

Se espera que se diviertan y aprendan.

2. CONTEXTO

La ciudad de Nueva York está sufriendo constantemente los ataques armados a los ciudadanos, especialmente por pandillas pequeñas de la ciudad. Al realizar una investigación, por parte de la policía, en especial por la capitana del cuerpo de policía de Nueva York, Yuriko Watanabe¹, se ha encontrado con muchos indicios que asocian a estas bandas con Wilson Grant Fisk², un empresario, dueño de las empresas Fisk, también conocido como Kingpin.

La empresa constructora de Fisk, ha estado construyendo y reestructurando algunos edificios en la ciudad y es en estos lugares donde Watanabe sospecha que se están entregando las armas a las pandillas locales. El problema es que ella no puede ingresar directamente, dado que Fisk posee demasiados contactos, y si se entera que existe una orden para allanar sus edificios, el logrará esconder la evidencia y no será llevado a la cárcel, por esta razón, Yuriko solicita la ayuda a uno de los super héroes locales, Spiderman³, para que pueda encontrar con las manos en la masa a Kingpin.

Spiderman es un superhéroe quién no mágicamente conseguirá solucionar el problema que posee Yuri, dado que para lograr encontrar la evidencia necesaria para atrapar a Fisk, debe saber cómo recorrer todos los edificios que estén involucrados de forma eficiente, con tal de no dejar puntos ciegos a la detective Watanabe.

En este punto es dónde Spiderman le solicita ayuda al director del Departamento de Ingeniería Informática (DIINF) de la Universidad de Santiago de Chile, dado que sabe que de allí salen las y los mejores analistas y programadores del mundo, por lo que podrán solucionar el problema de realizar la ruta más efectiva para poder inculpar de forma correcta a Kingpin, y por fin meterlo en la cárcel. Como los alumnos del DIINF son tan capaces de hacer esto, el director traspasa el problema a Pablo Román, coordinador del primer curso de las carreras del DIINF, quien solicita a los y las docentes de Métodos de Programación que hagan llegar el problema a sus estudiantes, para que lo puedan resolver.

Dado que sus estudiantes son excelentes en todo lo que hacen, Pablo solicita que la solución no solo sea aplicable a Nueva York, sino que a cualquier ciudad en dónde pueda ocurrir algo parecido, y así tener una solución general.

Es el momento en que usted, junto a un compañero del curso, pueda convertirse en superhéroe, ayudar a Spiderman y también al mundo. Para esto deberá simular el cómo recorrer una ciudad, en específico, los edificios que han sido marcados como puntos de contrabando de armas, entregando la mejor ruta y así Spiderman podrá dar las pruebas a la detective Watanabe.

¹ Personaje de ficción creado por Dan Slott y John Romita Jr.

² Personaje de ficción creado por Stan Lee y John Romita Jr.

³ Personaje de ficción creado por Stan Lee, Steve Ditko y Jack Kirby.

3. CARACTERÍSTICAS DE LA SIMULACIÓN

Para la simulación solicitada, a usted se le entregará una estructura de la ciudad, pero por temas de seguridad, esta será entregada de forma oculta. Las ciudades siempre serán representadas por una matriz cuadrada de tamaño $N \times N$, y los edificios de Kingpin serán una celda de esta matriz. Para ubicar estos edificios, solo se le entregará a usted un archivo de entrada, llamado “Entrada.in”, en donde en las primeras N líneas se indica la cantidad de edificios consecutivos que existen en cada una de las filas de la representación de la ciudad. Para saber cuál es la distribución de los edificios en las columnas de la representación de la ciudad, las edificaciones están en las siguientes N líneas del archivo de entrada, en donde los edificios consecutivos están separados por un guión.

Pero Taskmaster⁴, un villano aliado de Kingpin, ha hallado la forma en que le hace entrega la información a usted, por lo que también le ha estado enviado archivos que poseen la mismas características, por lo que Yuri ha añadido al archivo una información extra, la cual cantidad de veces que se deberá revisar el edificio de la ciudad, ya que se ha visto que a pesar que se ataque una vez el edificio, y se dejen a los pandilleros de Fisk atados para que la policía los pueda capturar, llega una nueva ola de pandilleros que los desata. Esta información se añade a la línea que indica la cantidad de edificios consecutivos de la ciudad, cosa que Taskmaster no sabe aún.

De esta forma, por ejemplo, si la ciudad a representar es la mostrada en la Figura 3.1, el documento que a usted le llegará está se muestra en la Figura 3.2.

		Columnas									
Filas		1	2	3	4	5	6	7	8	9	10
	1			1	2	1	1				
	2				2	1					
	3			1	2						
	4				1	1	1	2	1		
	5						1	1			
	6				1	2	1				
	7					1					
	8			1	2	2					
	9				1	3	3	2			
	10					2	2				

Figura 3.1: “Ciudad a representar”

⁴ Personaje de ficción creado por David Michelinie y George Pérez.

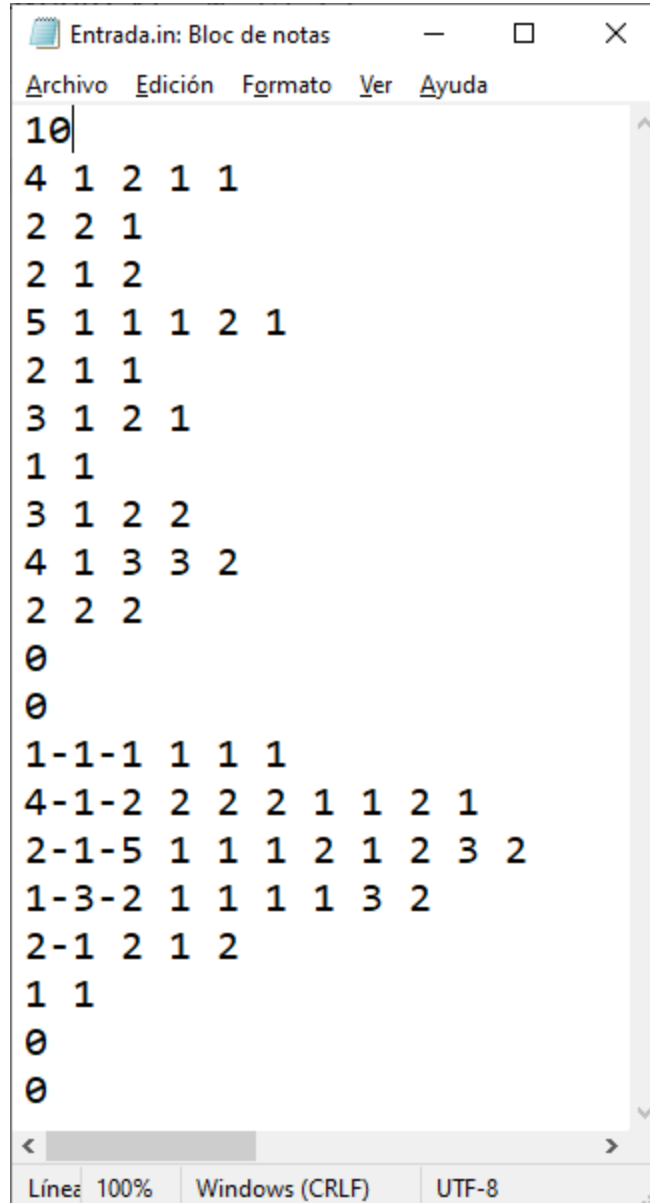


Figura 3.2: “Ejemplo de entrada para la ciudad de la Figura 3.1”

La idea, es que, a partir de un archivo similar al de la Figura 3.2, usted pueda informar a Spiderman la forma que posee la ciudad, específicamente, debe indicar en un archivo de salida, llamado “**primeraSalida.out**”, la cantidad de edificios de Fisk, su ubicación y las veces que se tiene que revisar. En la Figura 3.3 se muestra un ejemplo del archivo de salida, acorde al ejemplo dado.

Además, Spiderman necesita saber cómo debe recorrer la ciudad, con el fin de que recorra todos los edificios de Fisk las veces necesarias, devolviendo un archivo de texto, llamado “**segundaSalida.out**”, el cual debe tener los movimientos que debe hacer

Spiderman. Para esto usted deberá indicar con las siguientes letras los movimientos a realizar, separadas por un espacio:

- O (oeste) si Spiderman se debe mover a la izquierda del mapa.
- E (este) si Spiderman se debe mover a la derecha del mapa.
- N (norte) si Spiderman se debe mover arriba del mapa.
- S (sur) si Spiderman se debe mover abajo del mapa.

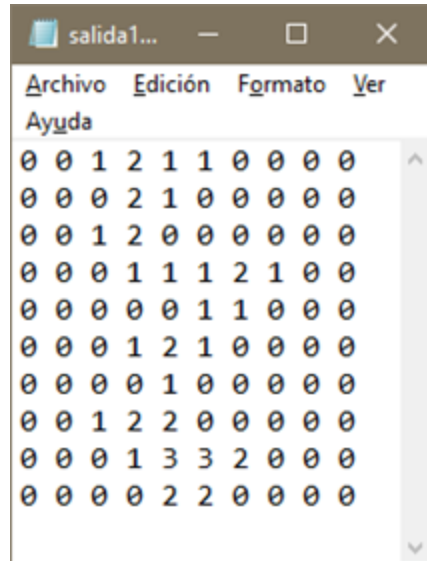


Figura 3.3: Ejemplo de salida **primeraSalida.out** para el ejemplo de la Figura 3.1

Para la realización del recorrido de Spiderman tome que cada vez que él se detiene en un edificio, realizará una revisión en sentido horario de cuál es el siguiente edificio a revisar, comenzando a mirar siempre al norte, por lo que sus movimientos poseen prioridad norte, oeste, este, sur. En la Figura 3.4 se muestra un ejemplo de salida que puede tener el mapa. Es importante indicar que no siempre existe un solo camino, puede haber varios, pero es necesario mostrar solo uno. Es probable también la configuración de la ciudad impida que Spiderman recorra todos los edificios, para lo cual el archivo “**segundaSalida.out**”, debe tener el mensaje: “No es posible recorrer todos los edificios de la ciudad”.



Figura 3.4: Segunda salida **segundaSalida.out** para la entrada de la Figura 3.1.

En los momentos en que Spiderman se encuentre con tiempo disponible, le gustaría también intentar resolver los mapas de la ciudad, para ver qué forma puede recorrerla,

para esto, Spiderman ingresará primero un lugar donde comenzará a recorrer la ciudad, el cual puede ser cualquier lugar que se encuentre en un borde de ésta.

Para esta parte, se debe mostrar a Spiderman la ciudad por pantalla, en especial debe mostrar la cantidad de edificios de Fisk que debe revisar. Posteriormente, por la entrada estándar debe ingresar uno de los movimientos válidos:

- O (oeste) si Spiderman se debe mover a la izquierda del mapa.
- E (este) si Spiderman se debe mover a la derecha del mapa.
- N (norte) si Spiderman se debe mover arriba del mapa.
- S (sur) si Spiderman se debe mover abajo del mapa.

Cada vez que pase por un edificio de Fisk, se descontará una revisión al edificio.

Tenga en cuenta, que tanto la solución como los movimientos de Spiderman siempre se realizan sobre los edificios de Fisk, y no se toman en cuenta aquellos en donde no existe presencia de este villano, por lo cual no deben ser tocados.

La jefa del Departamento de Policía de Nueva York, hay veces que no puede generar el archivo a enviar a Spiderman, por lo que también necesita de su ayuda para que ella, a partir de un archivo que representa la ciudad con la cantidad de edificios de Fisk en cada posición, se debe generar el archivo a hacer llegar a Spiderman. Es decir, se debe hacer el proceso inverso, donde se entregará un archivo como el de la Figura 3.3, con la diferencia que en la primera línea se indicará el tamaño de la ciudad $N \times N$, y se debe generar el archivo de la Figura 3.2, donde se representa la ciudad de la Figura 3.1. El nombre de este archivo siempre será **Entrada2.in**.

4. TRABAJO A REALIZAR

El trabajo a realizar se centra en distintos elementos a entregar para la evaluación de los conceptos aprendidos y aplicados. para eso se hacen ciertas acotaciones tanto de entradas y salidas como de funcionalidades que se deben llevar a cabo.

4.1. CARACTERÍSTICAS DE ENTRADA Y SALIDA

Para el desarrollo del problema se encuentran un archivo de entrada (Figura 3.2) y dos archivos de salida (Figura 3.3 y 3.4 respectivamente). En principal se tiene la siguiente explicación para la entrada mostrada:

- La primera línea (10): Indica el tamaño de la matriz. Esta siempre será cuadrada y los valores serán entre 5 y 10.
- Las 10 siguientes indican la cantidad de edificios contiguos que posee la fila en cuestión y las veces que se debe pasar por ese edificio en concreto, por ejemplo, para la fila 1 hay 4 edificios contiguos, en los cuales se debe pasar por ellos 1, 2, 1

y 1 vez respectivamente. En la tercera fila del archivo, se indican los edificios de la segunda fila de la ciudad, en este caso, 2 edificios, donde se deben recorrer 2 y 1 vez cada uno respectivamente.

- Las 10 siguientes líneas del archivo indican la distribución de las columnas en la ciudad.. En este caso, las dos primeras columnas no poseen valor, pero la tercera posee 3 edificios no contiguos entre ellos, donde por cada uno se debe pasar una sola vez. En la siguiente línea, existen 7 edificios en total para esa columna, donde hay 4 seguidos, un espacio o más, un edificio, un espacio o más, y dos edificios contiguos nuevamente. Antes de los 4 edificios contiguos o después de los dos finales, puede o no haber espacios en blanco.
- Se deberá considerar que el archivo de entrada siempre será correcto y no tendrá ningún problema de estructura según lo indicado en este documento.
- El nombre de este archivo siempre será **Entrada.in**.

En cuanto a las salidas del programa, se puede considerar que la salida señalada en la Figura 3.4 es una salida intermedia que servirá de entrada para generar la salida de la Figura 3.5.

En la Figura 3.4 podemos ver la representación de la ciudad completa ya armada, indicando los edificios existentes con números positivos mientras que donde no hay edificios se marca con un 0. Para generar esto se utiliza principalmente los primeros valores del archivo de entrada, siendo el primer valor para las filas, mientras que los primeros N elementos concatenados con un “-” para las columnas.

La Figura 3.5 es el recorrido que se debe hacer por cada edificio representado en la Figura 3.4, por lo cual esta última será utilizada también como entrada al problema. El recorrido debe ser mostrado como se indica en la sección 3 de este documento.

También desde un archivo parecido al de la Figura 3.3, se debe generar la salida de la construcción de la ciudad, como en el ejemplo de la Figura 4.1 mostrado a continuación.

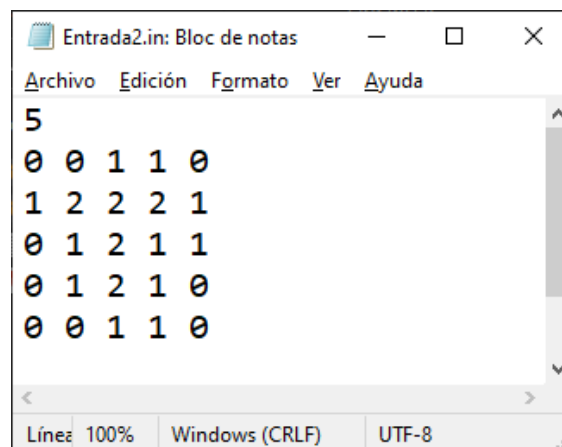
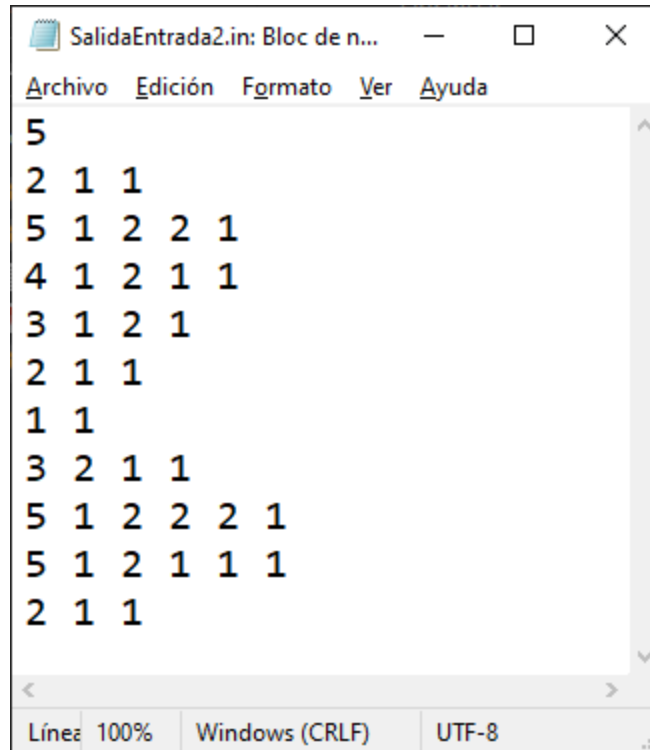


Figura 4.1: Archivo similar a **primeraSalida.out** utilizado para generar un archivo de entrada nuevo.

De esta entrada el programa tendría que entregar la salida de l a Figura 4.2..



```
5
2 1 1
5 1 2 2 1
4 1 2 1 1
3 1 2 1
2 1 1
1 1
3 2 1 1
5 1 2 2 2 1
5 1 2 1 1 1
2 1 1
```

Figura 4.2: Salida de la entrada de la Figura 4.1

4.2. FUNCIONALIDADES

Para el desarrollo de la simulación a realizar por su equipo de trabajo, se solicita cumplir con las siguientes características de la implementación y funcionalidades, las cuales son:

1. Se debe construir la solución al problema presentado utilizando el lenguaje de programación C.
2. La implementación debe leer los archivos de entrada⁵, acorde a lo señalado en el enunciado con los mismos nombres.
3. La implementación debe entregar los archivos de salida⁶, acorde a lo señalado en el enunciado.
4. Se debe construir la ciudad acorde a las reglas señaladas en el presente documento y el archivo leído.

⁵ Corresponden a Entrada.in y entrada2.in

⁶ Corresponden a primeraSalida.out y segundaSalida.out.

5. Se debe entregar las reglas de la construcción de la ciudad (**primeraSalida,out**) a partir de la ciudad representada con la cantidad de edificios de Fisk y el tamaño de la ciudad (**Entrada.in**).
6. Se debe permitir que se realice un recorrido automático de la ciudad, es decir, debe implementar un algoritmo el cual permita que Spiderman recorra automáticamente la ciudad pasando las veces indicadas por cada edificio siguiendo las reglas de ella, para esto la implementación deberá ubicar a Spiderman siempre en una posición del nororiente⁷ del mapa para recorrerlo de forma automática.
7. La implementación debe permitir al usuario moverse por la ciudad respetando las reglas de movimiento, es decir, el programa deberá ubicar al usuario en la posición más al nororiente de la ciudad y este mediante instrucciones N, O, S, E, se deberá mover a través de ésta, y el programa deberá informar cuando el usuario pase más veces de las debidas por un edificio de Fisk o intente pasar por un lugar que no pueda hacerlo..
8. La implementación debe permitir reconocer el lugar en dónde se encuentra Spiderman al momento de recorrer la ciudad de forma manual.
9. La implementación, al hacer un recorrido de forma manual, debe permitir solo los movimientos válidos, y no deberá mostrar como opción de movimientos aquellos que no se pueden realizar.
10. La implementación debe cumplir con todas las reglas de movimiento señaladas en el presente documento.
11. La implementación debe ser capaz de recuperarse ante errores cometidos por el usuario, señalando el error cometido. Un ejemplo de este tipo de errores puede ser que el tamaño señalado de filas o columnas sean distintas a las existentes, o tratar de realizar un movimiento inválido.
12. La implementación debe realizar un registro en un archivo de texto sobre las acciones realizadas tanto por el usuario como por la máquina. Dicho archivo debe tener la siguiente información:
 - a. En el nombre debe indicar la fecha y hora de inicio de la partida, señalando como mínimo el día, mes, año, hora y minuto de inicio.
 - b. Debe registrar cada acción realizada por el dueño del registro (usuario o IA) mostrando la hora y la acción realizada.
 - c. En caso de mostrar el movimiento, posición en la que se encuentra actualmente y movimiento que realizó para llegar allí.
13. Se debe informar, tanto en el archivo de texto o en la salida estándar, según corresponda, si se puede o no recorrer toda la ciudad.

⁷ Se refiere a la posición superior a la izquierda de la ciudad. En el primer espacio donde exista un edificio.

14. En caso de que se esté recorriendo de forma manual, el programa debe solicitar un siguiente movimiento, siempre y cuando sea posible terminar de recorrer todos los edificios de Fisk, desde la posición actual.
15. Considere que el tamaño de la ciudad, es decir, el N mencionado anteriormente variará entre 5 y 10.

Dentro de la construcción se contempla la implementación de las siguientes funcionalidades especiales, que son opcionales a realizar por el equipo de trabajo. Estas son:

1. Debe existir la posibilidad de modificar el tamaño de la ciudad a valores mayores que 10, determinando que la generación de la ciudad no se debe demorar más de 5 minutos.
2. Debe existir la posibilidad de parar el recorrido de la ciudad, sin rendirse, y poder volver a tomarla en cualquier momento.

5. ENTREGAS Y EVALUACIÓN

La entrega final consta de una aplicación funcional y la documentación necesaria de esta. El sistema creado debe cumplir con todas las funcionalidades de forma completa (siendo un total de quince funcionalidades) a excepción de aquellas que estén nombradas como opcionales, las cuales significarán una nota extra para las entregas 3.2 y 4.

La entrega del proyecto está dividida por 4 hitos, los cuales poseen una ponderación de 10, 20 30 y 40% respectivamente. La división de las entregas está basada en la estructura de la construcción de un software, las cuales son el análisis del problema, el diseño de la solución y la construcción. La última etapa consta de la mejora del software de las fallas que este pueda tener.

Cada grupo deberá estar compuesto por **dos alumnos que pertenezcan a la misma sección del laboratorio**. No puede haber grupos compuestos por más integrantes, a menos que el profesor lo estime conveniente. Esto último involucra una mayor exigencia en cuanto a la calidad del trabajo presentado, pero no así en las funcionalidades a realizar. En caso de que un alumno no posea grupo, el profesor lo podrá designar a uno que él encuentre conveniente. En caso de que un alumno desee cambiar de grupo una vez iniciado el proyecto, deberá encontrar un alumno de otro grupo que esté de acuerdo con intercambiar de grupo, en dicho caso el profesor confirmará o rechazará el intercambio entre los grupos.

5.1 ENTREGA 1

La entrega N°1 consta de la etapa de análisis del problema y el inicio de la etapa de diseño, en donde el grupo de trabajo deberá hacer entrega de

Informe: El informe debe contener una portada, introducción, descripción del problema, descripción de la solución y conclusiones

Introducción: Debe tener un acercamiento para el lector hacia el problema a resolver, además será necesario que realice un marco teórico en el cual se explique los conceptos utilizados para resolver la problemática a desarrollar, dentro de la introducción también deberá indicar la estructura del informe y su organización.

Descripción del problema: Se debe explicar en qué consiste el problema general y el contexto de éste.

Descripción de la solución: Se debe explicar cómo se afrontará el problema y la resolución de éste, indicando el método de resolución del problema, su justificación explicando por qué se escoge este método. En caso de existir sub problemas se debe hacer el mismo proceso.

Conclusiones: Se debe indicar cuales son los resultados del problema y que es lo que queda por realizar para resolver el problema general.

Anexo: Como anexo deben entregar el documento relleno correspondiente al desarrollo del trabajo dentro del grupo y los tiempos destinados a éste.

Además, el informe deberá contener un índice de contenidos, figuras y tablas.

El principal punto que evaluar dentro de este hito es verificar el entendimiento del problema por parte del grupo.

Las entregas de los documentos correspondientes a esta entrega deben ser subidos en el sitio de CampusVirtual el **viernes 07 de mayo del 2021 antes de las 23:55 horas**.

La evaluación de esta etapa está dividida en:

Informe: Tiene un 100% de ponderación para esta entrega. Este tendrá un 10% en formato, 20% ortografía y redacción, y 70% fondo del informe. El informe no debe tener más de 15 páginas de extensión, sin contar los anexos.

Con respecto a **la entrega mínima** de esta etapa corresponde a un informe que abarque todos los contenidos mencionados anteriormente.

5.2 ENTREGA 2

Para la entrega N°2, correspondiente al diseño el grupo deberá hacer entrega de una presentación que contenga:

Introducción: Debe tener un acercamiento para la audiencia hacia el problema a los temas tratados dentro de la presentación, haciendo hincapié en los métodos de resolución de problemas, dentro de la introducción también deberá indicar la estructura de la presentación y su organización.

Descripción del problema: Se debe explicar en qué consiste el problema general y el contexto de éste. Se debe indicar la forma de resolución de este, indicando el método de resolución del problema, su justificación explicando por qué se escoge este método. En caso de existir sub problemas se debe hacer el mismo proceso.

Diseño de la aplicación: Se debe explicar cómo estará estructurada la aplicación de acuerdo con la modularidad que desee realizar el grupo para desarrollo final. En este punto, el equipo puede utilizar distintos tipos de diagramas para poder mostrar esta información.

Conclusiones: Se debe indicar cuáles han sido los resultados de esta etapa y que es lo que se espera que suceda en la próxima. También deben hacer saber acá cuáles son los posibles problemas que esperan en la etapa de desarrollo y cómo los piensan abordar.

Las imágenes deben ser legibles dentro de la presentación.

La presentación tendrá que ser un vídeo con una duración máxima de 15 minutos.

Las entregas de los documentos correspondientes a esta entrega deben ser subidos en el sitio de CampusVirtual el **viernes 04 de junio de 2021 antes de las 23:55 horas**.

La evaluación de esta etapa está dividida en:

Presentación: Tiene un 100%, la cual está dividida en un 30% en la forma y un 70% del fondo de la presentación.

Con respecto a **la entrega mínima** de esta etapa corresponde a una presentación en vídeo la cual posea cada uno de los ítems mencionados anteriormente.

5.3 ENTREGA 3

En cuanto a la entrega 3 se solicitarán esta vez dos elementos, los cuales son una presentación y la aplicación.

La presentación deberá tener como contenidos los siguientes puntos:

Introducción: Debe tener un acercamiento para el auditor hacia el problema a los temas tratados dentro de la presentación, haciendo hincapié en los métodos de resolución de problemas, dentro de la introducción también deberá indicar la estructura de la presentación y su organización.

Descripción del problema: Se debe explicar en qué consiste el problema general y el contexto de éste. Se debe indicar la forma de resolución de este, indicando el método de resolución del problema, su justificación explicando por qué se escoge este método. En caso de existir sub problemas se debe hacer el mismo proceso.

Diseño de la aplicación: Se debe explicar cómo estará estructurada la aplicación de acuerdo con la modularidad que desee realizar el grupo para desarrollo final. En este punto, el equipo puede utilizar distintos tipos de diagramas para poder mostrar esta información.

Descripción de la aplicación: Se debe mostrar cómo está estructurada la aplicación, además se deben establecer las comparaciones entre el diseño y la construcción final.

Conclusiones: Se debe indicar cuáles han sido los resultados de esta etapa y cuáles son los problemas que posee la aplicación vistos por el grupo de trabajo, además de analizar la aplicación de las estrategias de resolución de problemas en la aplicación, mostrando sus ventajas y desventajas. Otro punto que tendrán que abordar será la ejecución del código de acuerdo con las pruebas realizadas, donde se recomienda analizar sus resultados.

Las imágenes deben ser legibles dentro de la presentación.

El programa debe cumplir con lo siguiente:

- Una aplicación desarrollada en **C**.
- La aplicación debe estar escrita en el paradigma de **Programación imperativo procedural**.
- La aplicación debe cumplir con todas las funcionalidades mencionadas en la entrega general.
- La aplicación debe poseer un **control de errores** para las acciones del usuario.

En la presentación el grupo deberá mostrar **la aplicación funcionando**, la cual debe ser una fiel copia del archivo subido a la plataforma CampusVirtual

, deben mostrar la ejecución completa de dos funcionalidades, de los cuales uno será seleccionado por el grupo deberá escoger uno y informarlo antes de la presentación, mientras que el profesor, antes de la presentación escogerá otro. Al momento de la presentación de las funcionalidades el grupo debe mostrar cómo se comunica su aplicación internamente para cumplir con el objetivo. Esta presentación tendrá una duración determinada por cada uno de los profesores.

La entrega de los documentos correspondientes a esta entrega (presentación y código) deben ser subidos en el sitio de UdeSantiagoVirtual.

Esta entrega posee una entrega formativa, y otra evaluativa, en donde solo la segunda posee una nota, pero ambas deben ser entregadas y poseer los mismos elementos, solamente que en la formativa, se les indicará que necesita mayor profundidad para entregar en la evaluativa.

La entrega formativa se realizará **el viernes 21 de junio de 2021**.

La entrega evaluativa se realizará **el viernes 23 de julio de 2021**.

La evaluación de esta etapa está dividida en:

Presentación: Tiene un 40%, lo cual está dividido en un 30% en la forma y un 70% del fondo de la presentación.

Aplicación: Tiene un 60% de ponderación, la cual estará compuesta en un 10% en la correcta aplicación de la programación, lenguaje y el orden del código, un 20% de la modularidad del código y un 70% con respecto al funcionamiento de este.

Con respecto a **la entrega mínima** de esta etapa corresponde a una presentación de la aplicación y la aplicación en sí la cual debe al menos compilar.

5.4 ENTREGA 4

La entrega 4 corresponde a la corrección completa de la entrega 3 de acuerdo con los errores indicados por el cuerpo docente.

La entrega de este Hito deberá hacerse por la plataforma de CampusVirtual **el 20 de agosto de 2021**.

Esta entrega posee como entrega la presentación de la aplicación solamente, sin ser necesario un documento de una presentación por separado, es decir, se puede solo mostrar la ejecución de la simulación realizada.

Con respecto a **la entrega mínima** de esta etapa corresponde a la aplicación en sí la cual debe al menos compilar.

5.5 ENTREGAS EN GENERAL

La no entrega de uno de los hitos (evaluativos y formativos), significará la nota mínima directamente para el proyecto.

Todas las entregas poseen una entrega mínima, cuyo incumplimiento significará la nota mínima en el hito. Además de lo indicado en cada uno de los ítems, cada grupo

deberá realizar la corrección de la entrega anterior, dejando de esta forma cada ítem con al menos una corrección por parte del profesor⁸.

En caso de existir atraso en la entrega, **se aplicará un descuento de 1 punto en la nota final del hito por cada hora de atraso**, a contar de 30 minutos después de la hora límite de subida.

En caso de que los link no estén disponibles o la plataforma no esté en funcionamiento, se les solicita enviar los elementos a los profesores y ayudantes correspondientes mediante un correo y subiendo el material a la plataforma LOA. **correspondiente**.

⁸ Queda a criterio del profesor si los ítems necesitan ser corregidos nuevamente, el cual también verá cómo lo evaluará.