UNIVERSIDAD CATOLICA DE COSTA RICA

INGIENERIA EN SISTEMAS

INTRODUCCION A LA PROGRAMACION

PROYECTO FINAL DIAGRAMA DE FLUJO EN RAPTOR

PROFESOR ANDRES JIMENEZ LEANDRO

ESTUDIANTE: JOSUE MUÑOZ SOLIS

SEDE SAN CARLOS, MARZO DE 2016

Tabla de contenido

[Introducción 4](#_Toc447396955)

[Realizar un diagrama de flujo 5](#_Toc447396956)

[Diagrama de flujo y sus características, problemas, soluciones del proyecto. 6](#_Toc447396957)

[Conclusión 7](#_Toc447396958)

[Cronograma y Bitácora de actividades 8](#_Toc447396959)

[Programa diagrama de flujo en raptor 8](#_Toc447396960)

[BIBLIOGRAFIA 9](#_Toc447396961)

[Anexos 9](#_Toc447396962)

# Introducción

Este trabajo es realizado para poner en práctica lo aprendido en clase, en la introducción a la programación se nos enseñó todo lo que tiene que ver con las bases de la programación desde lo más simple como reconocer definiciones importantes, generar soluciones, crear diagramas de flujo, reconocerlos, identificar cada signo de los diagramas, saber cómo se utilizan que solucionan., también la importancia de un pseudocodigo para partir desde el inicio que es escribir un programa a partir del pseudocodigo.

Por otro lado emplear la lógica que hemos ido desarrollando para llevar a cabo tares y trabajos que necesitan de un análisis de razonamiento, en el trabajo que les estoy presentando se puede observar que todas esta habilidades aprendidas en clase son aplicadas para formar un diagrama funcional y que llegue a solucionar este cálculo de notas que se determinan entre 1 y 20, y así despejar de la manera más clara la forma ideal, según mi lógica de cómo se puede resolver este diagrama a partir de un psuedocódigo brindado como guía.

# Realizar un diagrama de flujo

Se nos plantea un ejerció en el cual se nos brinda la información debida para comenzar un diagrama de flujo desde cero, para esto el profesor que es el facilitador del conocimiento, nos entrega como una base a seguir el siguiente pseudocodigo



Teniendo como base para iniciar este pseudocodigo, procedemos a realizarlo en un generador como herramienta de diagramas de flujo RAPTOR, para ello lo instalamos en nuestro pc en caso de no tenerlo, y proseguimos a analizar el pseudocodigo.

Utilizando esta herramienta debemos reconocer y utilizar correctamente los buques de decisión, terminales, acción, entrada, salida, procedimiento, los cuales nos ayudara a darle forma al diagrama de flujo, asignándole a cada uno la información correcta.

# Diagrama de flujo y sus características, problemas, soluciones del proyecto.

Como se mencionó anteriormente este diagrama se realizará desde cero en esta herramienta llamada RAPTOR., Pero uno de los grandes inconvenientes que se me presentaron en el desarrollo del diagrama fue la poca familiarización con dicha aplicación, si bien no es muy distinta a DFD., RAPTOR nos cambia un poco sobre lo que son los buques y nos obliga a tratar un poco más con él para entenderlo, al poco tiempo de utilizarlo podemos crear un diagrama de flujo tan sencillo como lo hacíamos en DFD, aparte de ese primer inconveniente también nos topamos con lo que es la interpretación del pseudocodigo ya que en lo personal al principio me costó un poco entenderlo y darle forma en RATOR, algunas cosas e información no me calzaba de la manera en la que yo quería y se me complico rearmar el diagrama de tal forma que la información se presentara con claridad al inicio. Una vez superado esto ingresando un output para que se mostrara las calificaciones que serían más adelante ordenadas de manera que correspondieran a su nota o calificación. Las utilizaciones de los buques de decisión generan un poco de confusión en el momento de hacer que la lógica del diagrama me calzara, que se cumpliera una y si no que me realizara lo siguiente a lo que llamamos (si es si haga lo siguiente y si es no realice lo otro) aunque nos parezca un poco intuitivo no lo es del todo porque necesita lo principal que es interpretar con nuestra propia lógica.

Hacer que el diagrama finalice, si era positivo el resultado no tomo mucho tiempo, pero hacer que el diagrama pasara, revisara o leyera todas las opciones si no se cumplía la primera, la segunda y así sucesivamente comprobara en que categoría de calificación se encontraba, esto me dio la idea que creo es la que la mayoría de los compañeros vamos a coincidir, la cual es ubicar la decisión en el no de la otra decisión.

Con estas y algunas de las formas que fui armando el diagrama me soluciono lo requerido que era la asignación de un número a una calificación, el mensaje al final establece cual es la partencia del número que ingreses de la misma manera muestra un mensaje de información advirtiendo si el número no se encuentra entre 1 y 20 que es lo establecido para este diagrama en específico.

# Conclusión

El error es muy frecuente cuando no aplicamos la lógica de manera correcta que nos resuelva el problema para el cual solucionar. Como recomendación les aconsejo a leerse detenidamente el pseudocodigo tratar de comprenderlo hasta el punto de familiarizarse con el de manera que lo recuerdes casi de memoria, esto porque cuando tienes en mente una estructura de una solución bien definida, es más fácil llevarla a cabo en un diagrama. Esto no implica que se vaya a tener uno que otro traspié o contratiempo a la hora de realizarlo.

El orden es un factor súper importante ya que esto no ayuda a depurar cualquier error que nos salte a la vista, también nos ayuda mejor a la compresión de una segunda persona., si bien estéticamente mi diagrama no se ve tan llamativo, si podemos cambiarle la orientación haciéndolo más vertical, pero esto es solamente cuestión de gustos nada más.

Saber identificar los buques de RAPTOR es esencial porque así sabemos cuál utilizar para ingresar los datos del programa y que funcione correctamente.

Lo aprendido por mi persona va más allá de lo teórico y lo práctico., creo que unos de los factores más importantes en este trabajo fue la dedicación y el empeño que puse a la hora de realizarlo, saber que cualquier persona es capaz de desarrollar la lógica sin ninguna dificultad, porque la lógica es algo que todos poseemos pero que pocos utilizamos debido a que hoy día muchas tareas se nos facilita por los grandes avances en la tecnología, más sin embargo el desarrollo metal que se tiene en este curso y al realizar este programa de ordenamiento de calificaciones es muy gratificante para seguir adelante.

# Cronograma y Bitácora de actividades

## Programa diagrama de flujo en raptor

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Fecha | Tiempo | Cambios |
| Recibí el proyecto | 01/03/2016 | 0 | 0 |
| Leí el proyecto | 04/03/2016 | 1 HORA | 0 |
| Descargue Raptor, sublime text 3 y github | 06/03/2016 | 1 HORA | 0 |
| Inicie el diagrama | 08/03/2016 | 3 HORAS | 2 |
| Inicie el pseudocodigo | 10/03/2016 | 2 HORAS | 1 |
| Busque información en internet | 11/03/2016 | 3 HORAS | 0 |
| Estudie algunos ejemplos | 12/03/2016 | 1 HORA | 0 |
| Hice cambios en el diagrama | 13/03/2016 | 2 HORAS | 5 |
| Hice cambios en el pseudocodigo | 15/03/2016 | 1 HORA | 5 |
| Inicien el documento escrito | 17/03/2016 | 2 HORAS | 3 |
| Entregue el avance en github | 20/03/2016 | 10 MINUTOS | 2 |
| Termine el diagrama de flujo y el (while do) | 24/03/2016 | 2HORAS | 0 |
| Termine el pseudocodigo y el (while do) | 26/03/2016 | 2 HORAS | 6 |
| Trabaje en el documento | 27/03/2016 | 2 HORAS | 1 |
| Termine el documento | 31/03/2016 | 4 HORAS | 20 |
| Entregue el trabajo | 03/04/2016 | 2 HORAS | 0 |

# BIBLIOGRAFIA

<http://www.aprenderaprogramar.es/index.php?option=com_content&view=article&id=314:ejemplos-y-ejercicios-resueltos-de-algoritmia-diagramas-de-flujo-y-pseudocodigo-calcular-media-cu00140a&catid=28:curso-bases-programacion-nivel-i&Itemid=59>

<http://www.slideshare.net/luismarlmg/diagrama-de-flujos-ejemplos>

<http://www.albertods.com/tecnologia/4eso/programacion/ejemplos_diagramas_flujo.pdf>

<https://es.wikipedia.org/wiki/Diagrama_de_flujo>

<http://www.mis-algoritmos.com/aprenda-a-crear-diagramas-de-flujo>

<http://arantxa.ii.uam.es/~jmmartinez/Pseudocodigo.htm>

<http://www.areatecnologia.com/informatica/pseudocodigo.html>

<http://informaticapc.com/teoria-de-la-programacion/ejemplos-algoritmos-pseudocodigo.php>

# Anexos



