Trabajo final de Introducción a la Programación

Tema: Raptor

Alumno: Luis Roberto Zúñiga Sánchez.

Profesor: Andrés Jiménez Leandro.

Segundo Cuatrimestre 2016

# Resumen ejecutivo:

En este proyecto veremos la función que tiene el diagrama creado en Raptor, el cual clasifica cierta nota en su respectiva categoría, por ejemplo, si está entre 20 y 19 la nota se categoriza en una calificación A, si está entre 18 y 16 será una B.

# Descripción:

Aquí encontraremos la parte teórica de este trabajo, conoceremos más sobre la aplicación y los métodos que se llevan a cabo paso a paso para llegar al resultado final. Vamos a encontrar un abstract con el resumen en inglés para que pueda ser entendido por usuarios sin conocimiento del español, también nos toparemos con los objetivos generales y específicos que llegaremos a lograr con la realización del trabajo y sin dejar de lado los obstáculos y limitaciones que encontramos a la hora de la realización que para corregir y poder avanzar tuvimos que analizar fuertemente el problema y encontrar una solución que testeamos varias veces.

La aplicación Raptor es bastante utilizada para la creación de diagramas la cual ayuda a los estudiantes a visualizar sus algoritmos fácilmente y lo mejor es que Raptor es distribuido libremente como un servicio a la comunidad educativa.

Este tipo de diagramas nos ayuda a resolver problemas fácilmente sin necesidad de hacer un esfuerzo mayor al de ir introduciendo datos, sin embargo, existen diagramas muy complejos los cuales llevan un mayor tiempo de dedicación y mayores testeos o pruebas antes de poder entregar el proyecto como concluido.

# Abstract:

Here we have a summary in English, which we will introduce what quremos achieve with this project.

Raptor is a programming environment based on flowcharts. With this project we want to draw the attention and interest of student programs falicitan creating algorithms and diagrams as they are very rich visual theme and we can see how step by step they are making our processes, which makes them very easy to understand and it is fun to see our results.

RAPTOR has six basic symbols, where each symbol represents a unique type of instruction. The basic symbols are shown at the right. The top four statement types, Assignment, Call, Input, and Output, are explained in this reading, the bottom two types, Selection and Loops, will be explained in a future reading.

INPUT: Allow the user to enter data. Each data value is stored in a variable.

PROCESSING: Change the value of a variable using some type of mathematical calculation.

PROCESSING: Execute a group of instructions defined in the named procedure. In some cases some of the procedure arguments (i.e., variables) will be changed by the procedure's instructions.

OUTPUT: save to a file the value of a variable.

The RAPTOR development environment, allows comments to be added to your program. Comments are used to explain some aspect of a program to a human reader, especially in places where the program code is complex and hard to understand. Comments mean othing to the computer and are not executed. However, if comments are done well, they can make program much easier to understand for a human reader.

The resulting comment can be moved in the RAPTOR window by dragging it, but you typically do not need to move the default location of a comment.

# Objetivo General:

Demostrar lo importante que son los ayudantes para construir diagramas de flujo utilizando específicamente el programa Raptor para de esta forma crear un interés en los compañeros para que utilicen dicho programa y se faciliten la creación de los diagramas a realizar en el futuro.

Dar a conocer la importancia que tienen los programas de control de versiones como lo es GitHub.

# Objetivo Específico:

Realizar una defensa del proyecto final en el cual se explicará paso a paso como se realizó un diagrama con el programa llamado Raptor, de esta manera se enseña lo útil que es efectuar un proceso de procedimientos y verlos ejecutarse uno por uno.

Efectuar una breve explicación de la importancia de GitHub y programas de control de versiones en general para crear la intriga en los alumnos y que amplíen su conocimiento hacia ellos ya que es muy importante conocer de este tema ya que no funcionará mucho como futuros ingenieros en sistemas.

# Introduction:

Este trabajo se crea a partir de un Proyecto propuesto por el profesor, en el cual se debe realizar dos tipos de diagramas en Raptor. El primer diagrama es creado para calificar las notas de los estudiantes en diferentes categorías: A, B, C, D y E, para ellos se le pide al usuario que ingrese una nota y dependiendo en cual rango se encuentre, así será agrupada. El segundo diagrama se crea de la misma manera, sin embargo, este segundo debe llevar lo que se conoce como ciclo While-Do, lo cual hace que el diagrama vuelva a pedir la nota si esta no cumple con algunas especificaciones o muestre el final del diagrama categorizando la nota en su debida agrupación.

http://raptor.martincarlisle.com/