Informe del proyecto.

Algoritmos y Programación II

Prof.: José Padrón

SISTEMA DE INSCRIPCIONES EFICIENTE (SIE).

1)**Especificaciones funcionales:**

- El informe consta de descripción de las funciones usadas, parámetros asumidos, funciones adicionales, descripción de la estructura del proyecto, conclusiones .

El objetivo principal del proyecto fue que cada registro funcionara de manera adecuada y que cada acción sobre el mismo se generara correctamente, de manera que se pueda usar el proyecto para actualizar datos.

1.2)El programa consta con la capacidad de añadir, modificar, eliminar y mostrar cada uno de los registros así como lo exige los requerimientos propuestos.

1.2.1)La función añadir de cada registro consiste en la creación de una nueva estructura de tipo registro (nodo), inserción de elementos para ese nodo, inicialización de las listas enlazadas a esa estructuras y finalmente se enlaza el nodo con los otros nodos de ese mismo tipo de estructuras.

1.2.2)La función modificar en cada registro consiste en primer lugar en buscar aquel registro a modificar, al encontrarlo se le pide al usuario el atributo a modificar; sobrescribiendo el nodo en cuestión.

1.2.3)La eliminación se fundamenta primero en buscar aquel registro a eliminar, luego la creación de un puntero temporal que se quedara en el nodo a eliminar, para luego borrarlo.

1.2.4)La impresión en pantalla se basa en encontrar aquellos registros a imprimir y luego se utilizan funciones como el “printf” para mostrar en pantalla.

1.3)Para la elaboración del proyecto se asume que el usuario ingresara en aquellos campos la información correspondiente, es decir la introducción de valores como números en los campos que lo requieran. También se asumió que en la creación de registros el usuario no duplicara alguno de los mismos.

Se realizó el procedimiento de mostrar todos los alumnos como funcionalidad adicional.

1.4)Conclusiones:

Los pro del diseño del proyecto, básicamente es que nos permite la creación, modificación , eliminación e impresión de registros en pocas líneas de código.

El contra del diseño es que debemos comprobar en ocasiones registros de la multilista de semestre para luego colocar en la multilista de alumnos.

//hay q poner q no logramos el choque de horario /cupo de sección aquí.

**2)Especificaciones de programación:**

2.1) Diseño conceptual de las estructuras de datos

En el siguiente dibujo se expresa como se usaron los nodos, como fueron enlazados formando listas de cada tipo.

ESTRUCTURA

MATERIAS INSCRITAS

SEMESTRE

PLANILLA

ALUMNOS

HORARIO

SECCIONES

MATERIA

EJEMPLO DE ESTRUCTURA.

SECCIONES

Materia: trigonometría

Código: 1234

Créditos: 4

Materia: matemáticas

Código: 123

Créditos: 5

MATERIA

Semestre: 1

máximo de créditos: 10

SEMESTRE

Código: 1245

Numero de sección: 003

Cedula del profesor: 951753

Nombre: Juancho pancho

Cupo: 30

Turo: vespertino

Código: 124

Numero de sección: 002

Cedula del profesor: 159753

Nombre: José padrón

Cupo: 26

Turo: Diurno

Día: Lunes

Hora inicial: 7am

Hora final: 9 am

Día: Martes

Hora inicial: 9am

Hora final: 11am

HORARIO

Día: Lunes

Hora inicial: 9am

Hora final: 11 am

Día: Martes

Hora inicial: 1pm

Hora final: 3pm

Numero de la planilla: 2

Código de la sección: 1245

Código de la materia: 1234

Numero de la planilla: 1

Código de la sección: 124

Código de la materia: 123

Cedula: 21706592

Código de la sección: 1245

Código de la materia: 1234

Numero de a planilla: 2

Cedula: 21706592

Código de la sección: 124

Código de la materia: 123

Numero de la planilla: 1

PLANILLA

Materias inscritas

Cedula: 21706592

Nombre: Marco

Apellido: Lozano

Fecha de nacimiento: 14/04/1994

Créditos otorgados: 2

Índice académico: 18

Numero de expediente: 2

ALUMNOS

Como se puede contemplar en la imagen superior, se estructuró el proyecto mediante 2 multi-listas, una multi-lista que empieza con listas de tipo Semestre con su información respectiva (número y número de créditos), enlazada a Materias que posee nombre de la materia, créditos y NRC; de Materias enlaza a Secciones(de esa materia y contiene información como código y número de la sección) y por ultimo de Secciones enlazadas a Horario. La segunda multi-lista pertenece a el proceso de los alumnos, la lista comienza en Alumnos donde se guardan todos los datos referentes a los mismos (como nombre, cedula, apellido, etc.), listas de tipo Planilla enlazadas a Alumnos que contienen los códigos de las secciones y materias que el Alumno inscribirá; luego listas de tipo Materias Inscritas enlazadas a Planilla, las cuales contienen la misma información que en la planilla pero ya inscribe la materia y asigna al alumno en la sección y materia.

2.2) **Main:**

* Objetivo: Mostrar el menú principal del programa y convocación de los procesos requeridos por el usuario mediante la selección de opciones generadas por el programa.
* Descripción de los parámetros de entrada: se inicializa la variable op de tipo integer en -1(op hace referencia a la palabra opción), para que entre en el ciclo “while(op!=0)” este ciclo es el inicial y posee el menú del programa.
* Descripción de los parámetros de salida: mediante inserción del usuario la variable op será utilizada para elegir entre las distintas opciones, en pocas palabras op cambiara y si el usuario ingresa “0” saldrá del programa ya que interrumpe el ciclo inicial.
* Algoritmo:

Int opc=-1,while(op!=0), printf() de todas las opciones y decoraciones(Sistema de Inscripción de la UCAB por ejemplo), scanf() de la variable op ,hasta que op=0.

Agregar Nodo Principal de multi-lista:

* Objetivo: Añadir uno de los nodos principales a la multi-lista para poder enlazar, los nodos principales serian los iniciales de ambas multilistas, es decir, Semestre y Alumnos.
* Descripción de los parámetros de entrada: por referencia serian (semestre \*\*p) y (alumnos \*\*q) respectivamente en sus multilistas, estos punteros están inicializados en el Main, aparte se crea una variable auxiliar de tipo lista(referente al nodo) para los 2 procesos.
* Descripción de las salidas: nodo llenado con la información para cada estructura, enlazado a la lista ya existente.
* Algoritmo: lista \*aux=new lista, printf() de los datos a llenar, seguidos por scanf( , &aux->info) almacenando la información en el nodo, aux->sig=\*p/\*q (enlaza por cabeza el nodo a la estructura)y luego aux->aba=NULL (inicializando la lista que le procede), por ultimo (\*p/\*q)=aux.

Agregar Nodos Secundarios:

* Objetivo: Añadir nodos
* Descripción de los parámetros de entrada: por referencia serían los mismos de los Nodos Principales, se crea al encontrar la lista deseada mediante los nodos principales una variable auxiliar del tipo de la lista a crear con un nodo.
* Descripción de las salidas: nodo llenado con la información para cada estructura enlazado a las listas correspondientes.
* Algoritmo: lista \*aux=new lista,variable de tipo int, printf() para buscar el numero o código de lista a crear (pueden existir mas de 1), scanf() del numero ingresado, seguido de un ciclo while(aux!=int)aux=aux->sig, y un if(aux->elemento==numero); que si encuentra la lista le crea un nodo enlazado secundario \*x=new secundario, printf() y scanf() de los datos e información, llenando el nodo, seguido del enlace del nodo respectivo.

Sistema de inscripción:

* Objetivo: Inscribir al alumno en la materia y sección que quiera cursar.
* Descripción de los parámetros de entrada: se utilizó las listas Planilla y Materias Inscritas, cada una con los parámetros de referencia de los nodos secundarios escritos anteriormente, variable de tipo int num y ced inicializadas y un nodo de tipo Materia Inscrita.
* Descripción de los parámetros de salida: nodo de tipo Materia Inscrita que posee la información de la materia.
* Algoritmo: creación de un nodo secundario de tipo Panilla enlazado con Materia Inscrita, printf() y scanf() de los datos a ingresar para poder inscribir la materia, estos serian :

Numero de cedula del alumno y numero de la planilla(ya la planilla posee la información de la materia y son ubicadas por números, es decir cada planilla tiene un numero y una materia.

While(aux1){(if aux1->cedula==ced){planilla \*aux2 while(aux2){if aux2->numero==num

Aux2=aux2->sig}aux1=aux1->sig} en pocas palabras busca al alumno con la cedula y el numero de la planilla. Luego

si encuentra los datos mateinscritas \*l=new mateinscritas y se llena la información en el nodo mediante printf() y scanf(), aux2->aba=l (enlaza el nodo) y termina el proceso.

Modificación y Eliminación de Registros:

* Objetivo: modificar o eliminar alguno de los registros.
* Descripción de los parámetros de entrada: mismos parámetros de agregar nodos pero en eliminar se uso una variable auxiliar (temp).
* Descripción de los parámetros de salida: nodo modificado o eliminado delete temp.
* Algoritmo:

1. Modificación: while(aux){if (aux->elemento == valor), printf() de los campos a cambiar y scanf() del campo que se cambia.
2. Eliminación: while(aux) if(aux->elemento==valor) hasta encontrar el registro, al encontrarlo, temp=aux->sig; aux->sig=temp->sig; delete temp.

*Realizado por:*

Iván Moreno

Manuel Higueras

Marco Lozano