

Proyecto Final.

Carmona Ayala Mariana Zoe, Castillo Martinez Diego Leonardo, García Cerda Gerardo Daniel, Lechuga Castillo Shareny Ixchel, Sánchez Medina José Santiago.

Facultad de Ingeniería, UNAM.

Ciudad de México, México.

Zocava2016@gmail.com

leomartinez17y@gmail.com

daniel654garcia@gmail.com

shareny.lechug@gmail.com

josesantiago201116@gmail.com

Resumen: Este documento es parte de un proyecto final por parte de la Universidad Nacional Autónoma de México de la materia de Programación Orientada a Objetos impartida por Ayala Barbosa Jose Antonio para la evaluación de los conocimientos adquiridos durante el curso. Este documento es la creación de un programa de gestión escolar, la cual permite guardar registros de alumnos para generar y asignar números de inscripción de acuerdo al algoritmo de la Facultad de Ingeniería de la UNAM.

Palabras clave: MVC, BubbleSort, recursividad.

I. INTRODUCCIÓN

Uno de los trabajos de los programadores es la creación de programas que ayuden a las personas en un área en específico, ya sea en escuelas, oficinas o incluso comida, en nuestro caso en el área escolar.

La cantidad de alumnos que están inscritos en la Facultad de Ingeniería es de cientos alumnos y guardar la información de cada uno de ellos es una tarea difícil, y lo es aún más el llevar un control de cada uno de acuerdo a su carrera y las materias que han o no han cursado desde que entraron.

Los programadores hacen que este registro sea más sencillo mediante la creación de programas donde guardan cada uno de estos datos de forma ordenada y de fácil acceso, con ayuda de funciones y librerías que están al alcance dependiendo del lenguaje que se utilice.

Gracias a la creación de este programa se podrá replicar de manera sencilla lo que nuestra facultad realiza con nuestros datos mientras nos encontramos cursando nuestras carreras

II. DESARROLLO

A. Patrón MVC

Como se mencionó en la introducción, este proyecto sigue el patrón MVC, sus siglas significan Modelo - Vista - Controlador (Model - View - Controller, en inglés), este consiste en una arquitectura que utiliza sistemas donde se requiere el uso de interfaces de usuario. [1]

Separa el código en tres capas: modelos, vistas y controladores, tal como su nombre lo indica. En los modelos se trabaja con los datos, las vistas representan el estado del modelo y los controladores actúan como intermediarios entre el usuario y el sistema.

B. Generar de alumno

Para poder generar a los alumnos se crearon tres listas: la primera con nombres de hombres, la segunda con nombres de mujeres y la última para apellidos. Cada dato se recuperaba de un archivo txt de la página del INEGI, donde se encontraban nombres y apellidos.

Aleatoriamente se elegía un número, para así poder extraerlo del archivo txt y usarlo como nombre, asignando a la misma variable.

C. Generar dirección

Al generar las direcciones de los alumnos, como en el caso de generar alumno, se recuperaban datos de un archivo txt de la página del INEGI y se volvió a elegir un número al azar, el cual era el índice de donde se agarra la lista.

D. Generar número de cuenta

Para la generación de números de cuenta se generaban tres números al azar: el primero era el primer dígito del número de cuenta (este va del 0 hasta el 9), el segundo eran otros dígitos del número de cuenta (abarcaba del 1000 al 9999) y el tercero eran los dígitos restantes del número de cuenta (igual abarcaba del 1000 al 9999).

Se hizo un casteo de tipo String para al final unir esos 3 números aleatorios y generar los números de cuenta de los alumnos. Este proceso se realizaba para disminuir y evitar que los números de cuenta que se generaban fueran iguales.

Así mismo, este recibía una lista de alumnos donde el último elemento de esa lista usaba recursividad. Dentro de la recursividad se recorren todos los números de cuenta de la lista, si se detectaba que un número era igual a otro, este volvía a repetir el proceso de generar un número de cuenta hasta que ningún número de toda la lista se repitiera.

La recursividad cuenta con 3 posibles casos: el primer caso es cuando no existieron problemas y ningún número de cuenta se repitió, el segundo caso es cuando existía número repetido, pero justo al final de la lista, y el tercer caso es cuando existe número repetido en cualquier punto de la lista.

E. Generar historial académico

Este en específico fue uno de los más complejos en desarrollar, dentro de este se generó una clase de tipo materia para recuperar datos de una materia (una lista de tipo String con todas las materias cursadas por el alumno hasta el momento).

Para hacerlo de una forma sencilla solamente se eligieron 6 ingenierías: ingeniería en computación, ingeniería mecánica, ingeniería aeroespacial, ingeniería industrial, ingeniería eléctrica electrónica e ingeniería ambiental, las carreras cuentan con diez semestres y cada semestre cuenta con cinco materias (se cuenta con un máximo de 50 materias). Así mismo, los primeros cinco semestres de todas las carreras eran los mismos (tronco común), además de que todas las materias valen seis créditos.

Lo primero que se realizó fue recuperar todas las materias que existen de un archivo txt, luego se generaba un número al azar del 1 al 6 el cual correspondía a cada una de las carreras (cada número corresponde al mismo orden en el que se mencionaron las carreras anteriormente). Dentro de estos números al azar se generaban seis casos donde, dentro de ellos, se volvía a generar números al azar del 1 al 10, esto para asignar el semestre del alumno. Al revisar el código, podemos observar que se empieza a realizar un ciclo.

Una condición importante que se debe de tomar en cuenta es que el alumno no puede tener materias reprobadas antes del semestre del que va cursando.

Se realizan dos ciclos: en el primero se calculan las materias que hay y a estas se le restan cinco (las cuales son las materias que se cursan en un semestre), luego se genera un número

aleatorio tipo float para asignar la calificación de la materia desde el 6 hasta el 10, repitiendo el proceso con todas las materias menos 5 (menos el semestre actual), el segundo ciclo se realiza con el último semestre en el que va cursando el alumno, realiza lo mismo que el primer ciclo pero con la diferencia de que la calificación puede ser ya reprobatoria. El índice donde se empieza a recuperar los datos de materias que lleva distinta cada alumno aumenta de 50 a 50. Al finalizar esto el alumno ya tiene historial académico, donde se sabe su semestre, su promedio y todas las materias dependiendo del semestre en el que va.

Cabe resaltar que el último ciclo contiene una condición, la cual es que si el número generado es menor a 6 el atributo de materias no acreditadas se le suma 1, esto nos ayuda para calcular la velocidad del alumno.

F. Velocidad

La velocidad utiliza el último ciclo del historial académico, donde ya sabemos la cantidad de materias no acreditadas por el alumno.

Además, se utiliza el algoritmo establecido por la Facultad de Ingeniería para calcular la velocidad del alumno, el cual consiste en que se dividen los créditos del alumno y los créditos desde el ingreso y su resultado se multiplica por 100.

Antes de avanzar un poco más, debemos de tener en cuenta que en nuestra escuela hipotética el alumno no debe de tener reprobados semestres anteriores, por lo que solo existe la posibilidad de que los alumnos tengan el semestre actual reprobado y se recuperen ese semestre.

G. Generar calificación

A base de la calificación de cada una de las materias que hay en materias, se promedian, se suma y se divide entre los semestre por 5.

H. Generar edad

Dentro de este se genera un número aleatorio y entra varias condiciones, esto dependiendo del semestre: la primera condición es cuando el alumno va en el semestre 1 a 2, por lo tanto se asigna la edad de 16 a 18 años, en la segunda condición se considera que el alumno va en el semestre 3 a 4, por lo tanto la edad asignada va a ser de 18 a 20 años, en la tercera condición son los semestres 5 a 6 donde el alumno tendrá la edad de 20 a 24 años, y en la última condición el alumno está cursando los semestres 8 a 10, donde su edad será de 25 a 27 años.

I. Indicador escolar

Al llegar a este punto recuperamos todos los datos anteriores y utilizando nuevamente el algoritmo establecido por la Facultad de Ingeniería generamos la escolaridad.

La escolaridad se obtiene dividiendo las asignaturas aprobadas en ordinario y las asignaturas inscritas en ordinario, este resultado se multiplica por 100. El alumno puede tener de ninguna hasta varias materias acreditadas en extraordinario.

Gracias al indicador escolar, podemos sacar el número de inscripción de los alumnos.

J. Número de inscripción

Se genera un arreglo con la cantidad de alumnos que existen, de este se recuperan todos los indicadores de todos los alumnos y se usa BubbleSort de manera ascendente donde va dando todos los datos, todo esto en la clase principal. Se genera un ciclo dentro de otro ciclo donde se genera una lista ordenada que recibe una lista desordenada de alumnos y un arreglo desordenado con todos los índices, los alumnos se ordenan con base al indicador y a su vez el arreglo desordenado.

Primero arregla todo el arreglo de índices y se genera un ciclo, empezando desde el último elemento hasta el primer elemento del arreglo ordenado, el índice empieza a buscar que índice escolar de la lista desordenada es igual a la del arreglo ordenado, el último es el indicador más alto, por lo tanto es el número de inscripción que será el primero.

K. Generación CSV

Aquí es donde se realiza la creación de un archivo donde se encuentran todos los datos de los alumnos: nombre, edad, carrera, semestre, indicador escolar, número de cuenta, promedio, número de materias no acreditadas, dirección y número de inscripción.

L. Generar menú principal

El menú principal se divide en 3 opciones, si el usuario es alumno, profesor o si solamente desea salir.

- Opción alumno: dentro de este se verifica si el número de cuenta ingresado es válido, si es así, se imprimen los datos del alumno, si su número de cuenta es incorrecto, cuenta con 3 intentos más para ingresar antes de que el programa termine.

- Opción profesor: dentro de este se verifica si la contraseña del profesor es correcta, si es así, se le lleva a un menú donde puede modificar datos de los alumnos, o si lo desea, eliminarlos, así mismo, si la contraseña es incorrecta, cuenta con 3 intentos más antes de que el programa finalice.
- Opción salir: dentro de este simplemente el programa termina su ejecución.

III. CONCLUSIONES

-Carmona Ayala Mariana Zoe:

A pesar de lo fácil que este proyecto se veía, cada vez que nos adentramos a las posibilidades que surgían al momento de programar las opciones, nos dimos cuenta que era mucho más complejo de lo que creíamos, con ayuda de nuestros conocimientos, pudimos completar a tiempo el proyecto. El proyecto me ayudó a entender de mejor manera la creación de archivos dentro del programa, a como editarlos y comprobar su existencia.

-Castillo Martinez Diego Leonardo:

Este proyecto fue un poco complicado, más en la cuestión de generar historial académico, debido a que hubo un punto en la realización del programa en la que al imprimir los números de cuenta, estos se imprimían de forma errónea o solamente ceros aparecían, pero después de varios intentos, se pudo solucionar el error. Considero que este proyecto me hizo darme cuenta de lo difícil que es crear programas de este estilo, pero también me hizo darme cuenta la importancia de nosotros los programadores.

-García Cerda Gerardo Daniel:

El comienzo del proyecto fue difícil en varios aspectos, debido a que varias veces los errores que encontrábamos en el código eran un poco complejos, al pasar las semanas nos fuimos adaptando mucho más y fuimos aplicando los conocimientos adquiridos durante el semestre. Lo que me llevo de este proyecto es mi gran esfuerzo a la hora de poder completar este programa y realizarlo de manera exitosa, en un futuro lo volveré a realizar para que no sea tan complejo, porque, debido al tiempo, varias cosas pudieron ser más cortas dentro del código.

-Lechuga Castillo Sharenly Ixchel:

Cada día fue un reto programar cada dato de los alumnos para que estos se guardarán de forma correcta en las listas, nos tardamos un poco en resolver más errores dentro del código

pero al final fue un éxito. Considero que este tipo de proyectos nos ayudan a darnos cuenta de todo lo que hemos aprendido durante el semestre.

-Sánchez Medina José Santiago.

Al ver el proyecto terminado, pude comprender que al final de todo, el proyecto me ayudó a pulir los conocimientos adquiridos e incluso a investigar mis dudas sobre ciertos temas, los errores me ayudaban a mejorar y comprender cada cosa que estaba haciendo de manera errónea. Este proyecto me fascinó aunque me agobió en ciertos momentos, pero el aprender fue lo que más me gustó.

IV. REFERENCIAS

[1] “Qué es MVC”. (2020, jul. 28). [Internet]. Disponible en <https://desarrolloweb.com/articulos/que-es-mvc.html>

[2]J. M. Aguilar, (2019, oct. 15). “¿Qué es el patrón MVC en programación y por qué es útil?”. [Internet]. Disponible en <https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx>

[3] “Números de inscripción”. (s.f.) [Internet]. Disponible en <http://escolar.ingenieria.unam.mx/asesoria/numero/>

[4] “Natalidad”. (2021). [Internet]. Disponible en <https://cuentame.inegi.org.mx/poblacion/natalidad.aspx?tema=P>