Universidad Nacional Autónoma De México

Facultad de Ingeniería

Materia: Computo Móvil

Profesor: German Santos Jaimes

Alumno: Montecillo Sandoval Jose Alejandro

No. Cuenta: 313080470

Actividad 1: Programa para registrar Alumnos

Semestre: 2021-1

Fecha de entrega: 28/11/202

**Planteamiento del problema.**

Desarrollar un programa completamente en Swift.

El programa debe capturar desde línea comandos datos de alumnos con las siguientes propiedades:

* Nombre (String)
* Edad (Int)
* Promedio (Double)
* ¿Cuenta con beca? (Boolean)

Se deben capturar como mínimo 10 alumnos.

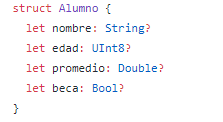
El programa debe presentar los siguientes resultados:

* Lista ordenada por Nombre
* Mostrar el alumno con promedio más bajo
* Mostrar el alumno con promedio más alto
* Mostrar solo los alumnos que cuentan con beca (lista ordenada por nombre)

**Propuesta de solución del problema.**

Se desarrollo el programa en Swift 5.3

Para poder obtener los resultados pedidos se creó un *struct* de Swift para poder agrupar los datos necesarios de cada alumno como se muestra a continuación.



Una vez teniendo el *struct* se crearon diferentes funciones que permitirán manejar el flujo de datos del programa, las cuales son:

* resgitrarAlumnos:

Esta función recibe el numero total de alumnos a registrar, el cual se le pregunta al usuario al iniciar el programa, ya con el número de alumnos estos se irán almacenando en un arreglo, y regresa una tupla del tipo *([Alumno], [Alumno])* esto quiere decir que regresa datos del tipo del *struct* creado al inicio y son dos listas una los alumnos ordenados alfabéticamente y otra ordenados por el valor de sus calificaciones.

* imprimirAlumnos:

Esta función recibe como parámetro la lista de alumnos ordenados alfabéticamente que regresa registrarAlumnos y los imprime.

* alumnosBeca:

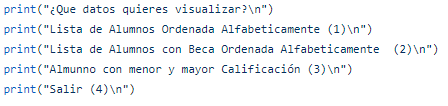
Esta función recibe como parámetro la lista de alumnos ordenados alfabéticamente que regresa registrarAlumnos y verifica si cuentan con una beca si es así los imprime si no, no los muestra.

* imprimirCalificaciones:

Esta función recibe como parámetro la lista de alumnos ordenados por calificación que regresa registrarAlumnos e imprime el primer y ultimo elemento de la lista que son el alumno con la calificación más baja y el alumno con la calificación más alta respectivamente

* main:

Esta función es la principal que se ejecuta y pregunta por la cantidad de alumnos a registrar debe ser un numero entero y este valor lo recibe como parámetro registrarAlumnos si se tiene un valor se ejecuta la función de resgitrarAlumnos si no se muestra un mensaje de error y termina el programa. Si se tienen datos se pedirán los datos de los alumnos que se quieren registrar, se almacenaran y se desplegara un menú con opciones para la visualización de los datos de la siguiente manera



Hasta que se presione la opción 4 de salir del programa este termina.

Para el ordenamiento de la lista de alumnos se hace uso de la función *sorted* propia del lenguaje Swift la que permite elegir porque elemento del *struct* se quieren ordenar los elementos.

Para el manejo de los opcionales que nos regresa la función readLine () para poder leer desde terminal se hace uso de diferentes técnicas, una es crear una variable auxiliar para hacer casteos después y otro es verificando que se tenga un valor con un if, haciendo en ambas el desenvolvimiento de la variable después para poder usarla.

**Capturas de pantalla del funcionamiento.**

Imagen 1. Número de alumnos a registrar y obtención de datos

Texto

Descripción generada automáticamente

Imagen 2. Menú

Texto

Descripción generada automáticamente

Imagen 3. Alumnos ordenados alfabéticamente

Texto

Descripción generada automáticamente

Imagen 4. Alumnos que cuentan con beca

Texto

Descripción generada automáticamente

Imagen 5. Alumnos con la calificación más baja y calificación más alta

Texto

Descripción generada automáticamente

**Código.**

|  |
| --- |
| struct Alumno { |
|  | let nombre: String? |
|  | let edad: UInt8? |
|  | let promedio: Double? |
|  | let beca: Bool? |
|  | } |
|  |  |
|  | func resgitrarAlumnos(alumnosRegistrar: Int) -> ([Alumno], [Alumno]){ |
|  | var ArrayAlumno = [Alumno]() |
|  | for index in 1...alumnosRegistrar{ |
|  | print("Datos alumno: ",index) |
|  | print("Nombre Alumno: " ) |
|  | let nombre = readLine()! |
|  | print("Edad Alumno: ") |
|  | let aux = readLine()! |
|  | let edad = UInt8(aux) |
|  | print("Promedia Alumno: ") |
|  | let aux2 = readLine()! |
|  | let promedio = Double(aux2) |
|  | print("¿Tiene Beca? 1:Si 2:No ") |
|  | var beca: Bool = true |
|  | let aux3 = readLine()! |
|  | let tieneBeca = UInt8(aux3) |
|  | if tieneBeca == 1 { |
|  | beca=true } |
|  | else if tieneBeca == 2{ |
|  | beca=false } |
|  |  |
|  | let alumno = Alumno(nombre:nombre,edad:edad,promedio:promedio,beca:beca) |
|  | ArrayAlumno.append(alumno) } |
|  |  |
|  | let alumnosOrdenados = ArrayAlumno.sorted{ |
|  | $0.nombre! < $1.nombre! } |
|  | let ordenadosCalificacion = ArrayAlumno.sorted{ |
|  | $0.promedio! < $1.promedio! } |
|  |  |
|  | return (alumnosOrdenados,ordenadosCalificacion) |
|  | } |
|  |  |
|  | func imprimirAlumnos(alumnosOrdenados: [Alumno]){ |
|  | print("Alumnos Ordenados alfabeticamente") |
|  | for alumno in alumnosOrdenados{ |
|  | print(alumno.nombre!) } |
|  | } |
|  |  |
|  | func alumnosBeca (alumnosOrdenados: [Alumno]) { |
|  | print("Alumnos con Beca: ") |
|  | for alumno in alumnosOrdenados{ |
|  | if alumno.beca == true { |
|  | print(alumno.nombre!)} } |
|  | } |
|  |  |
|  | func imprimirCalificaciones(ordenadosCalificacion: [Alumno]){ |
|  | print("Alumno con calificación mas baja \(ordenadosCalificacion[0].nombre!) : \(ordenadosCalificacion[0].promedio!)") |
|  | let ultimoElemento = ordenadosCalificacion.count - 1 |
|  | print("Alumno con calificación mas alta \(ordenadosCalificacion[ultimoElemento].nombre!) : \(ordenadosCalificacion[ultimoElemento].promedio!)") |
|  | } |
|  |  |
|  | func main() { |
|  | print("Programa para registro de Alumnos:\n") |
|  | print("Cantidad de alumnos a registrar: ") |
|  | if let alumnosRegistrar = Int(readLine()!){ |
|  | let ordenados = resgitrarAlumnos(alumnosRegistrar:alumnosRegistrar) |
|  | print("Se resgistraron \(alumnosRegistrar) alumnos correctamente\n") |
|  | print("¿Que datos quieres visualizar?\n") |
|  | print("Lista de Alumnos Ordenada Alfabeticamente (1)\n") |
|  | print("Lista de Alumnos con Beca Ordenada Alfabeticamente (2)\n") |
|  | print("Almunno con menor y mayor Calificación (3)\n") |
|  | print("Salir (4)\n") |
|  |  |
|  | while let opcion = Int(readLine()!) { |
|  | guard opcion != 4 else { |
|  | break |
|  | } |
|  | switch opcion { |
|  |  |
|  | case 1: |
|  | imprimirAlumnos(alumnosOrdenados: ordenados.0) |
|  |  |
|  | case 2: |
|  | alumnosBeca(alumnosOrdenados: ordenados.0) |
|  |  |
|  | case 3: |
|  | imprimirCalificaciones(ordenadosCalificacion: ordenados.1) |
|  | default: break |
|  | } |
|  |  |
|  | print("¿Que datos quieres visualizar?\n") |
|  | print("Lista de Alumnos Ordenada Alfabeticamente (1)\n") |
|  | print("Lista de Alumnos con Beca Ordenada Alfabeticamente (2)\n") |
|  | print("Almunno con menor y mayor Calificación (3)\n") |
|  | print("Salir (4)\n") |
|  | } |
|  | }else{ |
|  | print("No se ingreso un numero alumnos a registrar") } |
|  | } |
|  |  |
|  | //Funcion pincipal |
|  | main() |