Algoritmo BibliotecaProyecto

Definir Alumno como Registro

Nombre: Cadena[50]

Cedula: Cadena[15]

Fin Registro

Definir Libro como Registro

ID: Cadena[10]

Titulo: Cadena[100]

Categoria: Cadena[50]

Existencias: Entero

Fin Registro

Definir Prestamo como Registro

IDLibro: Cadena[10]

CedulaAlumno: Cadena[15]

DiaPrestamo: Entero

HoraPrestamo: Entero

MinutosPrestamo: Entero

Fin Registro

Definir ArchivoAlumnos, ArchivoLibros, ArchivoPrestamos como Archivo de Texto

Definir NuevoAlumno como Alumno

Definir NuevoLibro como Libro

Definir NuevoPrestamo como Prestamo

Definir Nombre, Cedula como Cadena

Funcion ExisteCedula(cedula: Cadena) Devolver Booleano

Definir ArchivoAlumnos como Archivo de Texto

Definir Linea como Cadena

Asignar 'alumnos.txt' a ArchivoAlumnos

Abrir ArchivoAlumnos

Leer ArchivoAlumnos, Linea

Mientras No EOF(ArchivoAlumnos) Hacer

Si Encontrar(Linea, 'Cedula: ' + cedula) > 0 Entonces

Cerrar ArchivoAlumnos

Devolver Verdadero

Fin Si

Leer ArchivoAlumnos, Linea

Fin Mientras

Cerrar ArchivoAlumnos

Devolver Falso

Fin Funcion

Funcion ExisteIDLibro(id: Cadena) Devolver Booleano

Definir ArchivoLibros como Archivo de Texto

Definir Linea como Cadena

Asignar 'libros.txt' a ArchivoLibros

Abrir ArchivoLibros

Leer ArchivoLibros, Linea

Mientras No EOF(ArchivoLibros) Hacer

Si Linea = id Entonces

Cerrar ArchivoLibros

Devolver Verdadero

Fin Si

Leer ArchivoLibros, Linea

Fin Mientras

Cerrar ArchivoLibros

Devolver Falso

Fin Funcion

Funcion PrestamoActivo(cedula, idLibro: Cadena) Devolver Booleano

Definir ArchivoPrestamos como Archivo de Texto

Definir Linea como Cadena

Definir Datos como Arreglo[1..2] de Cadena

Asignar 'prestamos.txt' a ArchivoPrestamos

Abrir ArchivoPrestamos

Mientras No EOF(ArchivoPrestamos) Hacer

Leer ArchivoPrestamos, Linea

Datos[1] <- Subcadena(Linea, Encontrar(Linea, ': ') + 2, Longitud(Linea))

Leer ArchivoPrestamos, Linea

Datos[2] <- Subcadena(Linea, Encontrar(Linea, ': ') + 2, Longitud(Linea))

Si (Datos[1] = cedula) Y (Datos[2] = idLibro) Entonces

Cerrar ArchivoPrestamos

Devolver Verdadero

Fin Si

Fin Mientras

Cerrar ArchivoPrestamos

Devolver Falso

Fin Funcion

Procedimiento AgregarAlumno(nombre, cedula: Cadena)

Definir ArchivoAlumnos como Archivo de Texto

Si ExisteCedula(cedula) Entonces

Escribir "La cedula ingresada ya existe. No se puede registrar otro alumno con la misma cedula."

Devolver

Fin Si

Si No ExisteArchivo('alumnos.txt') Entonces

ArchivoAlumnos <- Abrir("alumnos.txt")

Cerrar ArchivoAlumnos

Fin Si

ArchivoAlumnos <- Abrir("alumnos.txt")

Escribir ArchivoAlumnos, "Nombre: ", nombre, " - Cedula: ", cedula

Cerrar ArchivoAlumnos

Fin Procedimiento

Procedimiento AgregarLibro(id, titulo, categoria: Cadena; existencias: Entero)

Definir ArchivoLibros como Archivo de Texto

Si ExisteIDLibro(id) Entonces

Escribir "El ID del libro ingresado ya existe. Por favor, ingrese un ID único."

Devolver

Fin Si

Si No ExisteArchivo('libros.txt') Entonces

ArchivoLibros <- Abrir("libros.txt")

Cerrar ArchivoLibros

Fin Si

ArchivoLibros <- Abrir("libros.txt")

Escribir ArchivoLibros, id

Escribir ArchivoLibros, titulo

Escribir ArchivoLibros, categoria

Escribir ArchivoLibros, existencias

Cerrar ArchivoLibros

Fin Procedimiento

Procedimiento RegistrarPrestamo(cedula, idLibro: Cadena; diaPrestamo, horaPrestamo, minutosPrestamo: Entero)

Definir DatosLibro como Arreglo[1..4] de Cadena

Definir ArchivoLibros como Archivo de Texto

Definir ArchivoTemporal como Archivo de Texto

Definir LineaTemporal como Cadena

Definir Encontrado como Booleano

Si No ExisteCedula(cedula) Entonces

Escribir "La cedula del alumno ingresada no existe."

Devolver

Fin Si

Si No ExisteIDLibro(idLibro) Entonces

Escribir "El ID del libro ingresado no existe."

Devolver

Fin Si

Si ((horaPrestamo < 8) O (horaPrestamo > 16) O ((horaPrestamo = 8) Y (minutosPrestamo < 30)) O ((horaPrestamo = 16) Y (minutosPrestamo > 30))) Entonces

Escribir "Los prestamos solo pueden realizarse entre las 8:30 am y las 4:30 pm."

Devolver

Fin Si

Si ((diaPrestamo < 1) O (diaPrestamo > 5)) Entonces

Escribir "La biblioteca no trabaja los fines de semana. Los prestamos solo pueden hacerse de lunes a viernes."

Devolver

Fin Si

Abrir ArchivoLibros como "libros.txt"

Abrir ArchivoTemporal como "temporal.txt" para escritura

Encontrado <- Falso

Mientras No EOF(ArchivoLibros) Hacer

Leer ArchivoLibros, LineaTemporal

Si LineaTemporal = idLibro Entonces

Encontrado <- Verdadero

Leer ArchivoLibros, DatosLibro[1] // Leer titulo

Leer ArchivoLibros, DatosLibro[2] // Leer categora

Leer ArchivoLibros, DatosLibro[3] // Leer existencias

DatosLibro[4] <- ConvertirAString(ConvertirAEntero(DatosLibro[3]) - 1) // Restar una unidad al número de existencias

Escribir ArchivoTemporal, idLibro

Escribir ArchivoTemporal, DatosLibro[1]

Escribir ArchivoTemporal, DatosLibro[2]

Escribir ArchivoTemporal, DatosLibro[4]

Fin Si

Fin Mientras

Cerrar ArchivoLibros

Cerrar ArchivoTemporal

Borrar ArchivoLibros

Renombrar ArchivoTemporal como "libros.txt"

Si No Encontrado Entonces

Escribir "Error: No se encontro el libro con el ID especificado."

Devolver

Fin Si

Si No ExisteArchivo('prestamos.txt') Entonces

ArchivoPrestamos <- Abrir("prestamos.txt")

Cerrar ArchivoPrestamos

Fin Si

ArchivoPrestamos <- Abrir("prestamos.txt")

Escribir ArchivoPrestamos, "Cedula: ", cedula

Escribir ArchivoPrestamos, "ID Libro: ", idLibro

Escribir ArchivoPrestamos, "Dia del Prestamo: ", diaPrestamo

Escribir ArchivoPrestamos, "Hora del Prestamo: ", horaPrestamo, ":", minutosPrestamo

Cerrar ArchivoPrestamos

Escribir "Prestamo registrado con exito."

Fin Procedimiento

Procedimiento DevolverLibro(cedula, idLibro: Cadena)

Definir DatosLibro como Arreglo[1..4] de Cadena

Definir ArchivoLibros, ArchivoPrestamos, ArchivoTemporal como Archivo de Texto

Definir Linea, LineaTemporal como Cadena

Definir Encontrado como Booleano

Si No PrestamoActivo(cedula, idLibro) Entonces

Escribir "No hay un prestamo activo para el libro especificado y el alumno."

Devolver

Fin Si

Abrir ArchivoLibros como "libros.txt"

Abrir ArchivoTemporal como "temporal.txt" para escritura

Encontrado <- Falso

Mientras No EOF(ArchivoLibros) Hacer

Leer ArchivoLibros, Linea

Si Linea = idLibro Entonces

Encontrado <- Verdadero

Leer ArchivoLibros, DatosLibro[1] // Leer titulo

Leer ArchivoLibros, DatosLibro[2] // Leer categoria

Leer ArchivoLibros, DatosLibro[3] // Leer existencias

DatosLibro[4] <- ConvertirAString(ConvertirAEntero(DatosLibro[3]) + 1) // Sumar una unidad al número de existencias

Escribir ArchivoTemporal, idLibro

Escribir ArchivoTemporal, DatosLibro[1]

Escribir ArchivoTemporal, DatosLibro[2]

Escribir ArchivoTemporal, DatosLibro[4]

Fin Si

Fin Mientras

Cerrar ArchivoLibros

Cerrar ArchivoTemporal

Borrar ArchivoLibros

Renombrar ArchivoTemporal como "libros.txt"

Si No Encontrado Entonces

Escribir "Error: No se encontro el libro con el ID especificado."

Devolver

Fin Si

Abrir ArchivoPrestamos como "prestamos.txt"

Abrir ArchivoTemporal como "temporal.txt" para escritura

Mientras No EOF(ArchivoPrestamos) Hacer

Leer ArchivoPrestamos, Linea

Si (Subcadena(Linea, 1, 8) <> "Cedula: ") O (Subcadena(Linea, 10, Longitud(cedula)) <> cedula) Entonces

Escribir ArchivoTemporal, Linea

Sino

// Saltar las lineas correspondientes al prestamo

Leer ArchivoPrestamos, LineaTemporal

Leer ArchivoPrestamos, LineaTemporal

Leer ArchivoPrestamos, LineaTemporal

Leer ArchivoPrestamos, LineaTemporal

Fin Si

Fin Mientras

Cerrar ArchivoPrestamos

Cerrar ArchivoTemporal

Borrar ArchivoPrestamos

Renombrar ArchivoTemporal como "prestamos.txt"

Escribir "Libro devuelto con exito."

Fin Procedimiento

Procedimiento MostrarPrestamosActivos

Definir ArchivoPrestamos como Archivo de Texto

Definir Linea como Cadena

Abrir ArchivoPrestamos como "prestamos.txt"

Mientras No EOF(ArchivoPrestamos) Hacer

Leer ArchivoPrestamos, Linea

Escribir Linea

Fin Mientras

Cerrar ArchivoPrestamos

Fin Procedimiento

Procedimiento MostrarAlumnosDesdeArchivo

Definir ArchivoAlumnos como Archivo de Texto

Definir Linea como Cadena

Abrir ArchivoAlumnos como "alumnos.txt"

Mientras No EOF(ArchivoAlumnos) Hacer

Leer ArchivoAlumnos, Linea

Escribir Linea

Fin Mientras

Cerrar ArchivoAlumnos

Fin Procedimiento

Procedimiento MostrarLibrosDesdeArchivo

Definir ArchivoLibros como Archivo de Texto

Definir ID, Titulo, Categoria como Cadena

Definir Existencias como Entero

Abrir ArchivoLibros como "libros.txt" para lectura

Mientras No EOF(ArchivoLibros) Hacer

Leer ArchivoLibros, ID

Leer ArchivoLibros, Titulo

Leer ArchivoLibros, Categoria

Leer ArchivoLibros, Existencias

Escribir "ID: ", ID

Escribir "Titulo: ", Titulo

Escribir "Categoria: ", Categoria

Escribir "Existencias: ", Existencias

Escribir // Salto de línea para separar los libros

Fin Mientras

Cerrar ArchivoLibros

Fin Procedimiento

Procedimiento MostrarMenu

Escribir "--- Menu Principal ---"

Escribir "1. Agregar alumno"

Escribir "2. Agregar libro"

Escribir "3. Registrar prestamo"

Escribir "4. Devolver libro"

Escribir "5. Mostrar alumnos"

Escribir "6. Mostrar libros"

Escribir "7. Mostrar prestamos activos"

Escribir "8. Finalizar programa"

Escribir

Fin Procedimiento

Procedimiento EjecutarMenu

Definir Opcion como Caracter

Definir CedulaAlumno, IDLibro como Cadena

Definir DiaPrestamo, HoraPrestamo, MinutosPrestamo como Entero

Repetir

MostrarMenu

Escribir "Seleccione una opcion: "

Leer Opcion

Segun Opcion Hacer

'1':

Escribir "Ingrese el nombre del alumno: "

Leer Nombre

Escribir "Ingrese la cedula del alumno: "

Leer Cedula

AgregarAlumno(Nombre, Cedula)

'2':

Escribir "Ingrese el ID del libro: "

Leer NuevoLibro.ID

Escribir "Ingrese el titulo del libro: "

Leer NuevoLibro.Titulo

Escribir "Ingrese la categoria del libro: "

Leer NuevoLibro.Categoria

Escribir "Ingrese las existencias del libro: "

Leer NuevoLibro.Existencias

AgregarLibro(NuevoLibro.ID, NuevoLibro.Titulo, NuevoLibro.Categoria, NuevoLibro.Existencias)

'3':

Escribir "Ingrese la cedula del alumno: "

Leer CedulaAlumno

Si ExisteArchivo('alumnos.txt') Entonces

Escribir "Ingrese el ID del libro: "

Leer IDLibro

Si ExisteArchivo('libros.txt') Entonces

Escribir "Ingrese el dia de la semana del prestamo (1-5): "

Leer DiaPrestamo

Escribir "Ingrese la hora del prestamo (formato 24 horas): "

Leer HoraPrestamo

Repetir

Escribir "Ingrese los minutos del prestamo (0-59): "

Leer MinutosPrestamo

Hasta Que (MinutosPrestamo >= 0) y (MinutosPrestamo <= 59)

RegistrarPrestamo(CedulaAlumno, IDLibro, DiaPrestamo, HoraPrestamo, MinutosPrestamo)

Sino

Escribir "El ID del libro ingresado no existe."

Fin Si

Sino

Escribir "La cedula del alumno ingresada no existe."

Fin Si

'4':

Escribir "Ingrese la cedula del alumno: "

Leer CedulaAlumno

Escribir "Ingrese el ID del libro: "

Leer IDLibro

DevolverLibro(CedulaAlumno, IDLibro)

'5':

MostrarAlumnosDesdeArchivo

'6':

MostrarLibrosDesdeArchivo

'7':

MostrarPrestamosActivos

'8':

Escribir "Programa finalizado."

Otro:

Escribir "Opcion no valida. Por favor, seleccione nuevamente."

Fin Segun

Hasta Que Opcion = '8'

Fin Procedimiento

Inicio

EjecutarMenu

Fin