

INICIO

```
{ $codepage UTF8 }
```

```
usos crt,sysutils;
```

```
tipos
```

```
registro =registro
```

```
name: cadena;
```

```
lastname:cadena;
```

```
cedula: entero;
```

```
fin;
```

```
catalogo=registro
```

```
trabajos: cadena;
```

```
libros:cadena;
```

```
fin;
```

```
sacados = registro
```

```
name:cadena;
```

```
lastname:cadena;
```

```
prestados: cadena;
```

```
dias:real;
```

ID:entero;

fin;

lista = registro

name: cadena;

lastname: cadena;

dias: real;

prestado: cadena;

ID: entero;

fin;

variable

Horas, Minutos: Entero;

valname, valastname, valCI: booleano;

registros: arreglo [1..80] de registro;

Biblioteca: arreglo [1..40] de catalogo;

prestamo: arreglo [1..80] de sacados;

sanciones: arreglo [1..80] de lista;

procedimiento regisalum();

variable

a,m: entero;

txtFile:texto;

constante

i:entero=0;

INICIO

si i<80 entonces

inicio

i:=i+1;

repetir

Inicio

//El ingreso del nombre

Escribir linea('Ingrese su nombre:');

leer linea(registros[i].name);

limpiar pantalla;

valname:=verdadero;

si ((registros[i].name=") o (no(registros[i].name[1] en ['A'..'Z']))) entonces //Si la entrada es vacia o la primera letra no es mayuscula, no lo acepta.

```

        inicio
        valname:=falso;
        escribir linea('Datos del nombre no válidos, debes comenzar en mayúscula y evitar los números');
        delay(5000);
    fin
sino inicio
    para a:=2 a largo(registros[i].name) hacer
        inicio
        si no (registros[i].name[a] in ['a'..'z']) entonces //Si la entrada tiene numeros, caracteres especiales o una
mayuscula en el medio, no lo acepta.
            inicio
            valname:=falso;
            escribir linea('Datos del nombre no válidos, debes comenzar en mayúscula y evitar los números');
            delay(5000);
            romper; //Si consigue uno, no hace falta revisar el resa.
        fin;
    fin;
fin;
Fin;
limpiar pantalla;

```

```

hasta valname;
repetir
    Inicio
    escribir linea('Ingrese su apellido:');
    leer linea(registros[i].lastname);
    limpiar pantalla;
    valastname:=verdadero;
    si ((registros[i].lastname=") o (no(registros[i].lastname[1] en ['A'..'Z']))) entonces //Si la entrada es vacia o la primera letra
no es mayuscula, no lo acepta.
        inicio
        valastname:=falso;
        escribir linea('Datos del apellido no válidos, debes comenzar en mayúscula y evitar los números');
        delay(5000);
    fin
sino inicio
    para a:=2 a largo(registros[i].lastname) hacer
        inicio
        si no (registros[i].lastname[a] in ['a'..'z']) entonces //Si la entrada tiene numeros, caracteres especiales o una
mayuscula en el medio, no lo acepta.
            inicio

```

```

        valastname:=falso;
        escribir linea('Datos del apellido no válidos, debes comenzar en mayúscula y evitar los números');
        delay(5000);
        romper; //Si consigue uno, no hace falta revisar el res.
    fin;
fin;
fin;
Fin;
limpiar pantalla;
hasta valastname;
repetir
    escribir linea('Escriba su cédula de identidad');
    leer linea(registros[i].cedula);
    valCI:=verdadero;
    si (registros[i].cedula<1000000) o (registros[i].cedula>100000000)entonces
        Inicio
        valCI:=falso;
        escribir linea('Cédula de identidad falsa o caracter no permitido, vuelva a introducir su cédula');
        delay(5000);
    fin
fin

```

```
        limpiar pantalla;
    fin
sino
    Inicio
    escribir linea('Cédula de identidad aceptada');
    delay(2000);
    limpiar pantalla;
    Escribir linea('Ahora se esta haciendo una revisión de si es cédula existe no tardaremos...');
    delay(4000);
    limpiar pantalla;
    para m:=0 a i-1 hacer
        inicio
        si registros[m].cedula  $\neq$  registros[i].cedula entonces
            Inicio
            Escribir linea('espere...');
            delay(2000);
            limpiar pantalla;
        fin
    sino
```

Inicio

escribir linea('Cédula de identidad ya registrada, ingrese otra para poder registrase');

delay(5000);

limpiar pantalla;

romper;

fin;

fin;

fin;

hasta (valCI) y (registros[m].cedula <> registros[i].cedula) ;

escribir linea('Su registro ha sido completado');

escribir linea();

Escribir linea('Su ID siempre sera su número de cédula, si quiere pedir un préstamo de un libro o trabajo');

delay(5000);

limpiar pantalla;

fin

sino

Inicio

escribir linea('Nuestro registro esta lleno por el momento');

fin;



```
Asignar(txtFile, 'registros.txt');
si Archivo existe('registros.txt') entonces
    Adjuntar(txtFile)
sino
    Reescribir(txtFile);

si (registros[i].name <> "") entonces
    inicio
        escribir linea(txtFile, 'Nombre:', registros[i].name);
        escribir linea(txtFile, 'Apellido:', registros[i].lastname);
        escribir linea(txtFile, 'Cedula:', registros[i].cedula);
        escribir linea(txtFile, '-----');

    fin;

Cerrar(txtFile);

FIN;
```

procedimiento revislibtra(variable Biblioteca: arreglo de catalogo; prestamo: arreglo de sacados);

variable

i: entero;

inicio

Biblioteca[1].libros:='Frankenstein';

Biblioteca[2].libros:='Cien años de soledad';

Biblioteca[3].libros:='Los viajes de Gulliver';

Biblioteca[4].libros:='El Principia';

Biblioteca[5].libros:='Don Quijote de la Mancha';

Biblioteca[6].libros:='Alicia en el país de las maravillas';

Biblioteca[7].libros:='Moby Dick';

Biblioteca[8].libros:='Los miserables';

Biblioteca[9].libros:='Drácula';

Biblioteca[10].libros:='El código Da Vinci';

Biblioteca[11].libros:='Las crónicas de Narnia';

Biblioteca[12].libros:='La Biblia';

Biblioteca[13].libros:='La divina comedia';

Biblioteca[14].libros:='Crepúsculo';

Biblioteca[15].libros:='Fahrenheit 451';  
Biblioteca[16].libros:='Crónica de una muerte anunciada';  
Biblioteca[17].libros:='El juego de Ender';  
Biblioteca[18].libros:='El amo en los tiempos del cólera';  
Biblioteca[19].libros:='El gran Gatsby';  
Biblioteca[20].libros:='En busca del tiempo perdido';  
Biblioteca[1].trabajos:='El rol del mediador en la mediación judicial';  
Biblioteca[2].trabajos:='El juego para la enseñanza de las matemáticas';  
Biblioteca[3].trabajos:='El uso de hormigón reciclado en la construcción';  
Biblioteca[4].trabajos:='Los tratamientos efectivos para los ataques de pánico';  
Biblioteca[5].trabajos:='El rol fundamental del periodismo en el cuidado del medioambiente';  
Biblioteca[6].trabajos:='La crónica literaria como alternativa al periodismo tradicional';  
Biblioteca[7].trabajos:=' La recuperación de deportistas de alto rendimiento que han sufrido una lesión';  
Biblioteca[8].trabajos:='Análisis comparado de democracias federalistas';  
Biblioteca[9].trabajos:='Impresión 3D para fabricación de prótesis';  
Biblioteca[10].trabajos:='La comunicación horizontal y la comunicación vertical en relación con la eficiencia';  
Escribir linea('Tenemos los siguientes libros disponible');  
escribir linea('Presione cualquier tecla para ver la lista de trabajos');  
escribir linea;

```
para i := 0 a largo(Biblioteca)-1 hacer
    inicio

    si (Biblioteca[i].libros= ") entonces
        inicio

        fin
    sino
        inicio
        Escribir linea(i, ' ', Biblioteca[i].libros);
    fin;
fin;
leer linea();
limpiar pantalla;
Escribir linea('Tenemos los siguientes trabajos disponibles');
escribir linea('Presione cualquier tecla para escoger su libro/trabajo');
escribir linea;
para i := 0 a largo(Biblioteca)-1 hacer
    inicio
    si (Biblioteca[i].trabajos= ") entonces
```

inicio

fin

sino

inicio

Escribir linea(i, ' ', Biblioteca[i].trabajos);

fin;

fin;

leer linea();

limpiar pantalla;

fin;

procedimiento newbook(variable Biblioteca: arreglo de catalogo);

constante

l:entero=10;

i:entero=20;

variable

opcion: caracter;

INICIO

repetir

Inicio

Escribir linea('¿Quiere agregarlo como trabajo o libro?, si es libro presione "l" o si es trabajo presione "t"');

leer linea(opcion);

caso opcion de

'l': Inicio

si  $i < 40$  entonces

inicio

$i := i + 1$ ;

escribir linea('Aun tenemos espacio para almacenar su libro, escriba el nombre del libro');

leer linea(Biblioteca[i].libros);

limpiar pantalla;

Escribir linea('Ahora el libro ', Biblioteca[i].libros, ' esta en nuestra biblioteca disponible');

delay(2000);

Fin

sino

Inicio

Escribir linea('Nuestros archivos están llenos');

```

        fin;
    fin;
    't': Inicio
        si l<40 entonces
            Inicio
            l:= l + 1;
            escribir linea('Aun tenemos espacio para almacenar su trabajo, escriba el nombre del trabajo');
            leer linea(Biblioteca[l].trabajos);
            limpiar pantalla;
            Escribir linea('Ahora el trabajo ',Biblioteca[l].trabajos ,' esta en nuestro biblioteca disponible');
            delay(2000);
        Fin
    sino
        Inicio
        Escribir linea('Nuestros archivos estan llenos');
    fin;
    fin
    sino
        Inicio

```

```
        Escribir linea('opcion no existente');  
fin;  
fin;  
fin;  
hasta (opcion='t') o (opcion='T') o (opcion='l') o (opcion='L');  
FIN;
```

```
procedimiento prestalib (variable registros:arreglo de registro; sanciones: arreglo de lista);
```

```
    variable
```

```
    a,m,eleccion,posicion:entero;
```

```
    ID:entero;
```

```
    valID: booleano;
```

```
    opcion,opcion2: caracter;
```

```
    constante
```

```
    b:entero=0;
```



INICIO

si (largo(registros)> 0) entonces

Inicio

repetir

Inicio

escribir linea('Escriba su ID');

leer linea(ID);

valid:=verdadero;

si (ID<1000000) o (ID>100000000)entonces

Inicio

valid:=falso;

escribir linea('ID o caracter no permitido, vuelva a introducir su cédula debe encontrarse entre un  
millón cien millones');

delay(5000);

limpiar pantalla;

fin

sino

Inicio

```
        escribir linea('ID aceptada');
        delay(2000);
        limpiar pantalla;
    fin;
fin;
hasta valID;
para a:= 0 a largo(registros)-1 hacer
    inicio
    si (registros[a].cedula=ID) entonces
        inicio
        posicion:=a;
        Escribir linea('Bienvenido, presione cualquier tecla para continuar');
        escribir linea('Nombre: ',registros[posicion].name);
        escribir linea('Apellido: ',registros[posicion].lastname);
        escribir linea('ID: ',registros[posicion].cedula);
        leer linea;
        limpiar pantalla;
        revislibtra(Biblioteca, prestamo);
        b:=b+1;
```

repetir

Inicio

escribir linea('De lo mostrado, que quiere pedir prestado "L" para libros y "T" para trabajos');

leer linea(opcion2);

caso opcion2 de

'L': Inicio

Repetir

Inicio

eleccion:=0;

opcion:=' ';

Escribir linea('Escribe el número del libro de su elección');

leer linea(eleccion);

limpiar pantalla;

si Biblioteca[eleccion+1].libros =" entonces

Inicio

escribir linea('Libro no existente');

fin

sino

```

Inicio
para m:= 0 a b-1 hacer
    si prestamo[m].prestados = Biblioteca[eleccion+1].libros entonces
        inicio
            escribir linea('El libro ya está prestado');
            escribir linea('¿Desea elegir otro libro? (s/n)');
            leer linea(opcion);
            limpiar pantalla;
        fin
    sino
        inicio
            Escribir linea('Espere...');
            delay(1000);
            limpiar pantalla;
        fin;
    fin;
fin;
hasta (no(Biblioteca[eleccion+1].libros =)y no(opcion='s')) o (opcion='n') ;
fin;

```

'T':Inicio

Repetir

Inicio

eleccion:=0;

opcion:=' ';

Escribir linea('Escribe el número del trabajo de su elección');

leer linea(eleccion);

limpiar pantalla;

si Biblioteca[eleccion+1].trabajos =" entonces

Inicio

escribir linea('Trabajo no existente');

fin

sino

Inicio

para m:= 0 a b-1 hacer

si prestamo[m].prestados = Biblioteca[eleccion+1].trabajos entonces

inicio

escribir linea('El trabajo ya está prestado.');

escribir linea('¿Desea elegir otro trabajo? (s/n)');

```

                                leer linea(opcion);
                                limpiar pantalla;
                            fin
                        sino
                            inicio
                                Escribir linea('Espere...');
                                delay(1000);
                                limpiar pantalla;
                            fin;
                        fin;
                    hasta (no(Biblioteca[eleccion+1].trabajos =")y no(opcion='s')) o (opcion='n') ;
                Fin
            sino
                Inicio
                    escribir linea('Opción no existente');
                fin;
            fin;
        fin;
    
```

hasta (opcion2='T') o (opcion2 = 'L');

si opcion='n' entonces

Inicio

fin

sino

Inicio

si opcion2='L'entonces

Inicio

prestamo[b].name := registros[posicion].name;

prestamo[b].lastname:=registros[posicion].lastname;

prestamo[b].ID:=registros[posicion].cedula;

prestamo[b].dias:=3;

prestamo[b].prestados := Biblioteca[eleccion+1].libros;

escribir linea('Préstamo aprobado, tiene hasta tres días para devolverlo sino será sancionado');

delay(5000);

limpiar pantalla;

fin

sino

Inicio

prestamo[b].name := registros[posicion].name;

prestamo[b].lastname:=registros[posicion].lastname;

prestamo[b].ID:=registros[posicion].cedula;

prestamo[b].dias:=3;

prestamo[b].prestados := Biblioteca[eleccion+1].trabajos;

escribir linea('Préstamo aprobado, tiene hasta tres días para devolverlo sino será sancionado');

delay(5000);

limpiar pantalla;

fin;

fin;

Romper;

fin

sino

Inicio

Escribir linea('Espere...');

delay(200);

limpiar pantalla;

fin;



fin;

si (registros[a].cedula<>ID) entonces

Inicio

Escribir linea('ID no existente , vuelva a intentarlo, si no vaya a la opción de registro y registrese');

delay(5000);

limpiar pantalla;

fin

sino

Inicio

escribir linea();

fin;

fin;

FIN;

procedimiento renovacion ();

variable

a,posicion:entero;

ID:entero;

valID, renovar: booleano;

opcion: caracter;

INICIO

renovar:=falso;

si (largo(prestamo)> 0) entonces

Inicio

repetir

Inicio

escribir linea('Escriba su ID');

leer linea(ID);

valID:=verdadero;

si (ID<1000000) o (ID>1000000000)entonces

Inicio

valID:=falso;

cien millones');  
escribir linea('ID o caracter no permitido, vuelva a introducir su cédula debe encontrarse entre un millón

delay(5000);

limpiar pantalla;

fin

```

    sino
        Inicio
        escribir linea('ID aceptada');
        delay(2000);
        limpiar pantalla;
    fin;
fin;
hasta valid;
para a:= 0 a largo(prestamo)-1 hacer
    inicio
    si (prestamo[a].ID=ID) entonces
        inicio
        renovar:= verdadero;
        escribir linea('Bienvenido');
        posicion:=a;
        limpiar pantalla;
        escribir linea('Prestador: ', prestamo[posicion].name,' ',prestamo[posicion].lastname);
        escribir linea('Libro/Trabajo prestado: ', prestamo[posicion].prestados);
        escribir linea('Tiempo que le queda hasta su vencimiento: ', prestamo[posicion].dias:2:2);

```

```
escribir linea();  
repetir  
    escribir linea('Desea hacer una renovación de su préstamo?');  
    escribir linea('S para Si y N para No');  
    leer linea(opcion);  
    limpiar pantalla;  
    si (opcion='S') o (opcion='s') entonces  
        inicio  
        Escribir linea('Renovación realizada');  
        prestamo[posicion].dias:=3;  
        escribir linea('Prestador: ', prestamo[posicion].name,' ',prestamo[posicion].name);  
        escribir linea('Libro/Trabajo prestado: ', prestamo[posicion].prestados);  
        escribir linea('Tiempo que le queda hasta su vencimiento: ', prestamo[posicion].dias:2:2);  
        escribir linea('Presione cualquier tecla para continuar');  
        leer linea;  
        limpiar pantalla;  
    fin  
sino  
    inicio
```

```

        escribir linea('Cancelando renovación');
        escribir linea('Presione cualquier tecla para continuar');
        leer linea;
        limpiar pantalla;
    fin;
    hasta (opcion='S') o (opcion='s') o (opcion='n') o (opcion='N');
fin;
fin;
fin;
si no renovar entonces
    Inicio
        escribir linea('No hay nadie a quien renovar');
        delay(2000);
        limpiar pantalla;
    fin;
fin;

procedimiento actualizarPrestamos();
variable

```

```
f,num: entero;
```

```
    inicio
```

```
    num:=0;
```

```
// Verificar y procesar préstamos activos
```

```
si (largo(prestamo) > 0) entonces
```

```
    inicio
```

```
    para f := 0 a largo(prestamo)-1 hacer
```

```
        inicio
```

```
        // Solo procesar préstamos activos (no vacíos)
```

```
        si prestamo[f].name <> " entonces
```

```
            inicio
```

```
            prestamo[f].dias := prestamo[f].dias - 0.03;
```

```
            // Si los días restantes son menores o iguales a cero, se debe aplicar la sanción
```

```
            si prestamo[f].dias <= 0.00 entonces
```

```
                inicio
```

```
                num := num + 1;
```

```
                sanciones[num].name := prestamo[f].name;
```

```
                sanciones[num].lastname := prestamo[f].lastname;
```

```

    sanciones[num].ID := prestamo[f].ID;
    sanciones[num].prestado := prestamo[f].prestados;
    sanciones[num].dias := 7;
    escribir linea(sanciones[num].name, ' ', sanciones[num].lastname, ' Se le acabo el tiempo de préstamo, desde este momento
esta sancionado');
    prestamo[f].name := "";
    prestamo[f].lastname := "";
    prestamo[f].prestados := "";
    prestamo[f].ID := 0;
    prestamo[f].dias := -1;
    fin;
  fin;
fin;
fin;
fin;

```

```

procedimiento listactivos ();

```

```

variable

```

```

f,a: entero;

```

```

activos: booleano;

```

```

Prestados: texto;

    inicio

    activos := falso; // Variable para llevar un seguimiento de los préstamos activos


// Actualizar préstamos activos antes de listarlos
actualizarPrestamos();


// Verificar y procesar préstamos activos
si (largo(prestamo) > 0) entonces
    inicio
    para f := 0 a largo(prestamo)-1 hacer
        inicio

        // Solo procesar préstamos activos (no vacíos)
        si prestamo[f].name <> " entonces
            inicio

            activos := verdadero;

            Escribir linea('Nombre: ', prestamo[f].name, ' ', prestamo[f].lastname);
            escribir linea('Préstamo: ', prestamo[f].prestados);
            escribir linea('Tiempo restante: ', prestamo[f].dias:2:2);

```



```

leer linea();
limpiar pantalla;
Asignar(Prestados, 'Prestamos.txt');
si Archivo existe('Prestamos.txt') entonces
    Adjuntar(Prestados)
sino
    Reescribir(Prestados);
para a:= 0 a largo(prestamo)-1 hacer
    inicio
        si (prestamo[a].name <> "") entonces
            inicio
                Escribir linea(Prestados,'Nombre: ', prestamo[a].name, ' ',
prestamo[f].lastname);

                escribir linea(Prestados,'Préstamo: ', prestamo[a].prestados);
                escribir linea(Prestados,'Tiempo restante: ', prestamo[a].dias:2:2);
                escribir linea(Prestados, '-----');

            fin;
        fin;
    Cerrar(Prestados);

```

```
        fin;  
    fin;  
fin;
```

```
si no activos entonces
```

```
    inicio
```

```
        Escribir linea('No hay ningún préstamo activo');
```

```
        delay(2000);
```

```
        limpiar pantalla;
```

```
    fin;
```

```
fin;
```

```
procedimiento devoluciones;
```

```
variable
```

```
i, pos: entero;
```

```
ID: entero;
```

```
valID, activos: booleano;
```

inicio

activos := falso;

repetir

    inicio

    escribir linea('Escriba su ID');

    leer linea(ID);

    valID := verdadero;

    si (ID < 1000000) o (ID > 100000000) entonces

        inicio

        valID := falso;

        escribir linea('ID o caracter no permitido, vuelva a introducir su cédula debe encontrarse entre un  
millón cien millones');

        delay(5000);

        limpiar pantalla;

    fin

sino

    inicio

    escribir linea('ID aceptada');

    delay(2000);

```
        limpiar pantalla;
    fin;
fin;
hasta valID;

// Verificar si hay préstamos activos para la devolución
para i := 1 a Largo(prestamo) hacer
    inicio
        si prestamo[i].ID = ID entonces
            inicio
                pos := i;
                escribir linea('Bienvenido ');
                escribir linea('Prestador: ', prestamo[pos].name, ' ', prestamo[pos].lastname);
                escribir linea('Libro/Trabajo prestado: ', prestamo[pos].prestados);
                escribir linea('Se esta realizando la devolución...');
                prestamo[pos].name := "";
                prestamo[pos].lastname := "";
                prestamo[pos].prestados := "";
                prestamo[pos].dias := -1;
```

```

        prestamo[pos].ID := 0;
        activos := verdadero;
        romper; // Se encontró un préstamo, detener la búsqueda
    fin;
fin;

// Verificar si hay sanciones activas para la devolución
si no activos entonces
    inicio
    para i := 1 a Largo(sanciones) hacer
        inicio
        si sanciones[i].ID = ID entonces
            inicio
            pos := i;
            escribir linea('Bienvenido ');
            escribir linea('Sancionado: ', sanciones[pos].name, ' ', sanciones[pos].lastname);
            escribir linea('Libro/Trabajo sancionado: ', sanciones[pos].prestado);
            escribir linea('Se esta realizando la devolución...');
            sanciones[pos].prestado := "";

```

```

        activos := verdadero;
        romper; // Se encontró una sanción, detener la búsqueda
    fin;
fin;
fin;

si no activos entonces
    escribir linea('No hay préstamos ni sanciones para realizar una devolución');
    delay(4000);
    limpiar pantalla;
fin;

procedimiento actualizarSanciones();
variable
f,num: entero;

inicio
num:=0;
// Verificar y procesar préstamos activos

```

```

si (largo(sanciones) > 0) entonces
    inicio
    para f := 0 a largo(sanciones)-1 hacer
        inicio
        // Solo procesar préstamos activos (no vacíos)
        si sanciones[f].name <> " entonces
            inicio
            sanciones[f].dias := sanciones[f].dias - 0.03;
            // Si los días restantes son menores o iguales a cero, se debe aplicar la sanción
            si sanciones[f].dias <= 0.00 entonces
                inicio
                escribir linea(sanciones[num].name, ' ', sanciones[num].lastname, ' Se le acabo el tiempo de
préstamo, desde este momento esta sancionado');
                sanciones[f].name := "";
                sanciones[f].lastname := "";
                sanciones[f].ID := 0;
                sanciones[f].dias := -1;
            fin;
        fin;
    fin;
fin;

```

fin;

fin;

procedimiento listsancion ();

variable

f,a: entero;

sancionados: booleano;

penalizados:texto;

inicio

sancionados := falso; // Variable para llevar un seguimiento de las sanciones activas

actualizarSanciones;

si (largo(sanciones) > 0) entonces

inicio

para f := 1 a largo(sanciones)-1 hacer

inicio

// Solo procesar sanciones activas (no vacíos)



si sanciones[f].name <> " entonces

    inicio

        sancionados:=verdadero;

        Escribir linea('Nombre: ', sanciones[f].name, ' ', sanciones[f].lastname);

        escribir linea('Préstamo del sancionado: ', sanciones[f].prestado);

        escribir linea('Tiempo restante: ', sanciones[f].dias:2:2);

    leer linea();

    limpiar pantalla;

    Asignar(penalizados, 'penalizados.txt');

        si Archivo existe('penalizados.txt') entonces

            Adjuntar(penalizados)

        sino

            Reescribir(penalizados);

        para a := 1 a largo(sanciones)-1 hacer

            inicio

                si (sanciones[a].name <> ") entonces

                    inicio

                        Escribir    linea(penalizados,   'Nombre:   ',   sanciones[a].name,   '   ',

                        prestamo[f].lastname);

```

        escribir linea(penalizados,'Préstamo: ', sanciones[a].prestado);
    escribir linea(penalizados, 'Tiempo restante: ', sanciones[a].dias:2:2);
        escribir linea(penalizados, '-----');

        fin;
    fin;
    Cerrar(penalizados);
    fin;
    fin;
    fin;
    si no sancionados entonces
        inicio
            Escribir linea('No hay sanciones activas');
            delay(2000);
            limpiar pantalla;
        fin;
    fin;

```

procedimiento DiaSemana (variable Minutoactual:entero; Horaactual:entero);

constante

DiasSemana: arreglo[1..5] de cadena = ('Lunes','Martes','Miercoles','Jueves','Viernes');

p:entero=0;

Inicio

si (Horaactual=8) y (Minutoactual=30) entonces

    inicio

        p:=p+1;

        escribir linea(DiasSemana[p]);

    fin

sino

    inicio

        escribir linea(DiasSemana[p]);

    fin;

fin;

procedimiento AvanzarReloj();

    inicio

Minutos := Minutos + 15;

si Minutos >= 60 entonces

    inicio

        Minutos := Minutos - 60;

        Horas := Horas + 1;

        si (Horas >= 16) y (Minutos >=30) entonces

            inicio

                Horas := 8;

                Minutos:=15;

            fin;

        fin;

fin;

variable

opcion: caracter;

Horaactual, Minutoactual: Entero;

INICIO

Horas := 8;

Minutos := 15;

repetir

    inicio

        actualizarPrestamos;

        actualizarSanciones;

    si (Horas >= 16) y (Minutos >=30) entonces

        inicio

            Horas := 8;

            Minutos:=15;

    fin;

    AvanzarReloj;

Horaactual := Horas;

    Minutoactual := Minutos;

    DiaSemana(Minutoactual,Horaactual);

    escribir linea('Hora Actual: ', Horas, ':', Minutos);

    escribir linea('Escriba el número de la acción que quieres realizar');

    escribir linea();

    escribir linea('1. Registrarse');

```
    escribir linea('2. Sacar un libro');  
    Escribir linea('3. Revisar los libros disponibles');  
    Escribir linea('4. Ingresar un nuevo libros');  
    escribir linea('5. Revisar préstamos activos');  
    escribir linea('6. Revisar sanciones');  
    escribir linea('7. Realizar renovación');  
    escribir linea('8. Realizar devolución');  
    escribir linea('0. Salir');  
    escribir linea();  
    leer linea(opcion);  
    limpiar pantalla;  
    caso opcion de
```

```
'1': inicio
```

```
    regisalum;
```

```
    Fin;
```

```
'2': inicio
```

```
    prestalib(registros,sanciones);
```

```
fin;
```

```
'3': Inicio
```

```
        revislibtra (Biblioteca, prestamo);  
fin;  
'4': Inicio  
        newbook(Biblioteca);  
fin;  
'5': Inicio  
        listactivos();  
fin;  
'6': Inicio  
        listsancion();  
fin;  
'7': inicio  
        renovacion();  
fin;  
'8': inicio  
        devoluciones();  
fin;  
'0': Inicio  
        escribir linea('Se va a cerrar sesión');
```

fin

sino

Inicio

escribir linea('No existe esa opción');

leer linea();

limpiar pantalla;

Fin;

fin;

fin;

hasta (opcion ='0');

FIN.