

Proyecto: Penitenciaria Montaña Azul

Integrantes: Ricardo Espinosa y Juan Diego Guerrero

Análisis

Entrada	Proceso	Salida
"datosCarcel.csv" "Pabellon1.csv" "Pabellon2.csv" "Pabellon3.csv" "Pabellon4.csv" "Pabellon5.csv"	<ol style="list-style-type: none">1. Contraseña para editar los datos.2. Poder visualizar el archivo completo o un archivo por pabellón.3. Historia e información relevante sobre la penitenciaria/cárcel.4. Promedio de edades de toda la cárcel y promedio por pabellón.5. Edad más frecuente de toda la cárcel y por pabellón. (Moda).6. Delito más cometido según la clasificación.7. Búsqueda de PPL por su cedula.8. Búsqueda de PPLs por su celda.9. Registro de Personas Privadas de la Libertad.10. Generación de archivos para tratar los datos.11. Control de Visitas.12. Manejo de Agravantes.13. Generación de Estadísticas.	"datosCarcel.csv" "Pabellon1.csv" "Pabellon2.csv" "Pabellon3.csv" "Pabellon4.csv" "Pabellon5.csv" Estadística

Estructura – Análisis

CAPACIDAD MAXIMA: 800 PPL, 400 celdas, cada celda está diseñada para 2 PPL.

DIVISION EN PABELLONES DE LA ESTRUCTURA:

PABELLON 1 – MAXIMA SEGURIDAD (A): Alta peligrosidad

CAPACIDAD: 40 celdas – 80 PPL

PABELLON 2 (B): Mediana peligrosidad

CAPACIDAD: 90 celdas – 180 PPL

PABELLON 3 (C): Mediana peligrosidad

CAPACIDAD: 90 celdas – 180 PPL

PABELLON 4 (D): Baja peligrosidad

CAPACIDAD: 90 celdas – 180 PPL

PABELLON 5 (E): Baja peligrosidad

CAPACIDAD: 90 celdas – 180 PPL

Nota: Cada celda se enumera con la inicial del pabellón seguida del número.

Ejemplo:

Celda A03:

Corresponde a la celda 3 del pabellón 1 – máxima seguridad (A).

Celda D16:

Corresponde a la celda 16 del pabellón 4 – baja peligrosidad(D).

CLASIFICACION DE DELITOS:

ALTA PELIGROSIDAD -> (A):

Narcotráfico; Homicidio

MEDIANA PELIGROSIDAD -> (M):

Robo; Hurto; Fraude; Evasión

BAJA PELIGROSIDAD -> (B):

Invasión; Amenazas; Estafa; Otros

VISITAS:

ALTA PELIGROSIDAD -> (A):

TOTAL SEMANAL: 1 HORA

MEDIANA PELIGROSIDAD -> (M):

TOTAL SEMANAL: 2 HORAS

BAJA PELIGROSIDAD -> (B):

TOTAL SEMANAL: 4 HORAS

Historia -> Información de la penitenciaría/cárcel

La Penitenciaría Montaña Azul, ubicada en la ciudad de Loja, es una instalación penitenciaria que cumple con estándares de seguridad y disciplina. Con su arquitectura simple, el complejo consta de celdas básicas y un patio central para actividades controladas. Montaña Azul se enfoca en proporcionar un entorno seguro y ordenado para la reclusión, sin adornos innecesarios. Su presencia sólida representa un compromiso con la justicia y la contención de la delincuencia en Loja.

Diseño

Algoritmo presentarData

filas = 800

columnas = 13

nombreArchivoCompleto = "datoscarcel.csv"

pabellon1 = "pabellon1.csv"

pabellon2 = "pabellon2.csv"

pabellon3 = "pabellon3.csv"

pabellon4 = "pabellon4.csv"

pabellon5 = "pabellon5.csv"

Definir principal como matriz de cadenas de tamaño [filas][columnas]

generarNumeracion(principal, filas)

generarCedulas(principal, filas)

generarNombresPPL(principal, filas)

generarClasificacion(principal, filas)

generarFechasIngreso(principal, filas)

generaredades(principal, filas)

generarPenas(principal, filas)

generarFechasSalida(principal, filas)

generarAniosRestantes(principal, filas)

generarPabellon(principal, filas)

generarCelda(principal, filas)

generarVisita(principal, filas)

generardelitos(principal, filas)

generarcsvCompleto(principal, filas, nombreArchivoCompleto)

generarcsvPabellon1(principal, filas, pabellon1)

generarcsvPabellon2(principal, filas, pabellon2)

generarcsvPabellon3(principal, filas, pabellon3)

generarcsvPabellon4(principal, filas, pabellon4)

generarcsvPabellon5(principal, filas, pabellon5)

FinAlgoritmo

Función generarCSVCompleto(principal:s, filas: i, nombreArchivoCompleto: s): s

e = Escribir(nombreArchivoCompleto)

e = "#PPL; Cedula; Nombres; Edad(Anios); Pena(anios); Fecha(Ingreso)(d/m/a);
Fecha(Salida)(d/m/a); Anios Restantes (pena); Pabellon; Celda; Clasificacion;
Visitas semanales(Horas); Delito\n"

Para i desde 0 hasta filas hacer

e = principal[i][0], principal[i][1], principal[i][2], principal[i][3], principal[i][4],
principal[i][5], principal[i][6], principal[i][7], principal[i][8], principal[i][9],
principal[i][10], principal[i][11], principal[i][12])

FinPara

Finintentar

FinFunción

Función generarCSVPabellon1(principal: s, filas: i, pabellon1: s) : s

e = Escribir(pabellon1)

e = "#PPL; Cedula; Nombres; Edad(Anios); Pena(anios); Fecha(Ingreso)(d/m/a);
Fecha(Salida)(d/m/a); Anios Restantes (pena); Pabellon; Celda; Clasificacion;
Visitas semanales(Horas); Delito\n"

Para i desde 0 hasta 80 hacer

e = principal[i][0], principal[i][1], principal[i][2], principal[i][3], principal[i][4],
principal[i][5], principal[i][6], principal[i][7], principal[i][8], principal[i][9],
principal[i][10], principal[i][11], principal[i][12])

FinPara

Finintentar

FinFunción

Función generarCSVPabellon2(principal: s, filas: i, pabellon2: s) : s

e = Escribir(pabellon2)

e = "#PPL; Cedula; Nombres; Edad(Anios); Pena(anios); Fecha(Ingreso)(d/m/a);
Fecha(Salida)(d/m/a); Anios Restantes (pena); Pabellon; Celda; Clasificacion;
Visitas semanales(Horas); Delito\n"

Para i desde 80 hasta 260 hacer

e = principal[i][0], principal[i][1], principal[i][2], principal[i][3], principal[i][4],
principal[i][5], principal[i][6], principal[i][7], principal[i][8], principal[i][9],
principal[i][10], principal[i][11], principal[i][12])

FinPara

Finintentar

FinFunción

Función generarCSVPabellon3(principal: s, filas: i, pabellon3: s) : s

e = Escribir(pabellon3)

e = "#PPL; Cedula; Nombres; Edad(Anios); Pena(anios); Fecha(Ingreso)(d/m/a);
Fecha(Salida)(d/m/a); Anios Restantes (pena); Pabellon; Celda; Clasificacion;
Visitas semanales(Horas); Delito\n"

Para i desde 260 hasta filas 440 hacer

e = principal[i][0], principal[i][1], principal[i][2], principal[i][3], principal[i][4],
principal[i][5], principal[i][6], principal[i][7], principal[i][8], principal[i][9],
principal[i][10], principal[i][11], principal[i][12])

FinPara

Finintentar

FinFunción

Función generarCSVPabellon4(principal: s, filas: i, pabellon4: s) : s

e = Escribir(pabellon4)

e = "#PPL; Cedula; Nombres; Edad(Anios); Pena(anios); Fecha(Ingreso)(d/m/a);
Fecha(Salida)(d/m/a); Anios Restantes (pena); Pabellon; Celda; Clasificacion;
Visitas semanales(Horas); Delito\n"

Para i desde 440 hasta filas 620 hacer

e = principal[i][0], principal[i][1], principal[i][2], principal[i][3], principal[i][4],
principal[i][5], principal[i][6], principal[i][7], principal[i][8], principal[i][9],
principal[i][10], principal[i][11], principal[i][12])

FinPara

Finintentar

FinFunción

Función generarCSVPabellon5(principal: s, filas: i, pabellon5: s) : s

e = Escribir(pabellon5)

e = "#PPL; Cedula; Nombres; Edad(Anios); Pena(anios); Fecha(Ingreso)(d/m/a);
Fecha(Salida)(d/m/a); Anios Restantes (pena); Pabellon; Celda; Clasificacion;
Visitas semanales(Horas); Delito\n"

Para i desde 620 hasta filas 800 hacer

e = principal[i][0], principal[i][1], principal[i][2], principal[i][3], principal[i][4],
principal[i][5], principal[i][6], principal[i][7], principal[i][8], principal[i][9],
principal[i][10], principal[i][11], principal[i][12])

FinPara

Finintentar

FinFunción

Función generarNumeracion(principal: s, filas: i) : s

Para i desde 0 hasta filas hacer

principal[i][0] = "PPL ", (i + 1)

Fin Para

Fin Función

Función generarCedulas(principal: s, filas: i) : s

x1 = 0, x2 = 0

Para i desde 0 hasta longitud(principal) - 1 hacer

x1 = Azar(90102,10190)

x2 = Azar(90102,10190)

principal[i][1] = x1 + x2

Fin Para

Fin Función

Función generarNombresPPL(principal: s, filas: i) : s

nomPersonas1 = {"Roberto", "Dario", "Fabian", "Gonzalo", "Carlos", "Mario", "Pedro", "Victor",
"Michael", "Byron"}

nomPersonas2 = {"Antonio", "Cesar", "Pablo", "Adrian", "Diego", "Julian", "Sergio", "Paulo",
"Angel", "Vicente"}

apePersonas1 = {"Paredes", "Romero", "Cuenca", "Guerrero", "Garcia", "Torres", "Cevallos",
"Perero", "Gonzales", "Perez"}

apePersonas2 = {"Sanchez", "Ramirez", "Hidalgo", "Rivera", "Gomez", "Espinosa", "Reyes",
"Roman", "Morales", "Castro"}

Para i desde 0 hasta longitud(principal) hacer

indAleatorioNomb1 = Azar(nomPersonas1)

indAleatorioNomb2 = Azar (nomPersonas2)

indAleatorioApe1 = Azar (apePersonas1)

indAleatorioApe2 = Azar (apePersonas2)

principal[i][2] = nomPersonas1[indAleatorioNomb1],
nomPersonas2[indAleatorioNomb2], apePersonas1[indAleatorioApe1],
apePersonas2[indAleatorioApe2]

Fin Para

Fin Función

Función generaredades(principal: s, filas: i) : s

Edad = 0, anioActual = 0, anioIngreso = 0, resto = 0

fechaActual

anioActual

anioIngresoAux = nulo

aleatorio rand = Nuevo GeneradorAleatorio()

Para i desde 0 hasta filas hacer

 edad = Azar(18, 50)

 anioIngresoAux = Subcadena(principal[i][5], 7, Longitud(principal[i][5]))

 anioIngreso = anioIngresoAux

 resto = anioActual - anioIngreso

 Si edad <= (resto + 18) entonces

 edad = (resto + 18) + Azar(1, 5)

 Fin Si

 principal[i][3] = edad

Fin Para

Fin Función

Función generarPenas(principal: s, filas: i) : s

 Pena = 0, AnioIngreso = 0, resto = 0

 AnioIngresoaux = nulo

 fechaActual

 AnioActual

 Para i desde 0 hasta filas hacer

 pena = Azar(1, 30) // Pena máxima en Ecuador es 50 años

 AnioIngresoaux = Subcadena(principal[i][5], 7, Longitud(principal[i][5])) // String

 AnioIngreso = AnioIngresoaux // Entero

 resto = AnioActual - AnioIngreso

 Si pena <= resto entonces // Nunca van a salir penas incoherentes

 pena = resto + Azar(2, 7)

 Fin Si

 principal[i][4] = pena

 Fin Para

Fin Función

Función generarFechasIngreso(principal: s, filas: i) : s

día = 0, mes = 0, anio = 0

fechaActual

anioActual

Para i desde 0 hasta filas hacer

 dia = Azar(1, 31)

 mes = Azar (1, 12)

 anio = Azar (1980, anioActual)

 Si dia < 10 y mes < 10 entonces

 principal[i][5] = "0", día, "/" + mes, "/", anio

 Sino si dia < 10 entonces

 principal[i][5] = "0", día, "/", mes, "/", anio

 Sino si mes < 10 entonces

 principal[i][5] = día, "/" + mes, "/", anio

 Sino

 principal[i][5] = día, "/", mes, "/", anio

 Fin Si

Fin Para

Fin Función

Función generarFechasSalida(principal: s, filas: i) : s

 día = nulo, mes = nulo, anioIngreso = nulo, pena = nulo

 anioSalida = 0

 Para i desde 0 hasta filas hacer

 dia = Subcadena(principal[i][5], 1, 2)

 mes = Subcadena(principal[i][5], 4, 2)

 anioIngreso = Subcadena(principal[i][5], 7)

 pena = principal[i][4]

 anioSalida = anioIngreso + pena

 principal[i][6] = día, "/", mes, "/", anioSalida

 Fin Para

Fin Función

Función generarAniosRestantes(principal: s, filas: i) : s

 Aniosalida = nulo

anioActual = 0

fechaActual

anioActual

aniosRestantes = 0

Para i desde 0 hasta filas hacer

 aniosalida = Subcadena(principal[i][6], 7)

 aniosRestantes = aniosalida - anioActual

 principal[i][7] = aniosRestantes

Fin Para

Fin Función

Función generarClasificacion(principal: s, filas: i) : s

Para i desde 0 hasta filas hacer

 Si i >= 0 y i < 80 entonces

 principal[i][10] = "Alto"

 Sino si i >= 80 y i < 440 entonces

 principal[i][10] = "Medio"

 Sino si i >= 440 y i <= 799 entonces

 principal[i][10] = "Bajo"

 Fin Si

Fin Para

Fin Función

Función generarPabellon(principal: s, filas: i) : s

 medianaPeligrosidad = {"2(B)", "3(C)"}

 bajaPeligrosidad = {"4(D)", "5(E)"}

Para i desde 0 hasta filas hacer

 Si i >= 0 y i < 80 y principal[i][10] igual a "Alto" entonces

 principal[i][8] = "1(A)"

 Sino si i >= 80 y i < 260 y principal[i][10] igual a "Medio" entonces

 principal[i][8] = medianaPeligrosidad[0]

 Sino si i >= 260 y i < 440 y principal[i][10] igual a "Medio" entonces

 principal[i][8] = medianaPeligrosidad[1]

 Sino si i >= 440 y i < 620 y principal[i][10] igual a "Bajo" entonces

principal[i][8] = bajaPeligrosidad[0]

Sino si $i \geq 620$ y $i \leq 799$ y principal[i][10] igual a "Bajo" entonces

principal[i][8] = bajaPeligrosidad[1]

Fin Si

Fin Para

Fin Función

Función generarCelda(principal: s, filas: i) : s

cont = 0

Para i desde 0 hasta 39 hacer

principal[i][9] = "A" + (cont + 1)

cont = cont + 1

Fin Para

cont = 0

Para i desde 40 hasta 79 hacer

principal[i][9] = "A" + (cont + 1)

cont = cont + 1

Fin Para

Para i desde 80 hasta 169 hacer

principal[i][9] = "B" + (cont + 1)

cont = cont + 1

Fin Para

Para i desde 170 hasta 259 hacer

principal[i][9] = "B" + (cont + 1)

cont = cont + 1

Fin Para

Para i desde 260 hasta 349 hacer

principal[i][9] = "C" + (cont + 1)

cont = cont + 1

Fin Para

Para i desde 350 hasta 439 hacer

principal[i][9] = "C" + (cont + 1)

cont = cont + 1

Fin Para

Para i desde 440 hasta 529 hacer

principal[i][9] = "D" + (cont + 1)

cont = cont + 1

Fin Para

Para i desde 530 hasta 619 hacer

principal[i][9] = "D" + (cont + 1)

cont = cont + 1

Fin Para

Para i desde 620 hasta 709 hacer

principal[i][9] = "E" + (cont + 1)

cont = cont + 1

Fin Para

Para i desde 710 hasta 799 hacer

principal[i][9] = "E" + (cont + 1)

cont = cont + 1

Fin Para

Fin Función

Función generarVisita(principal: s, filas: i) : s

horas = 0

Para i desde 0 hasta filas hacer

Si principal[i][10] igual a "Alto" entonces

horas = 1

Sino si principal[i][10] igual a "Medio" entonces

horas = 2

Sino si principal[i][10] igual a "Bajo" entonces

horas = 4

Fin Si

principal[i][11] = horas, " Hora(s)"

Fin Para

Fin Función

Función generarDelitos(principal: s, filas: i) : s

Para i desde 0 hasta filas hacer

Si principal[i][10] igual a "Alto" entonces

delitos = {"Narcotrafico", "Homicidio"}

Sino si principal[i][10] igual a "Medio" entonces

delitos = {"Robo", "Hurto", "Fraude", "Evasion"}

Sino si principal[i][10] igual a "Bajo" entonces

delitos = {"Invasion", "Amenazas", "Estafa", "Otros"}

Fin Si

Fin Para

Fin Función

Fin

Prueba de escritorio

La informacion se guardo correctamente en: datoscarcel.csv

La informacion se guardo correctamente en: pabellon1.csv

La informacion se guardo correctamente en: pabellon2.csv

La informacion se guardo correctamente en: pabellon3.csv

La informacion se guardo correctamente en: pabellon4.csv

La informacion se guardo correctamente en: pabellon5.csv

Algoritmo ControlCarcelario

// Variables

archivoPrincipal = "datosCarcel.csv", archivoInfo = "informacionCarcel.txt", contraseña = null, pabellon1 = "pabellon1.csv", pabellon2 = "pabellon2.csv", pabellon3 = "pabellon3.csv", pabellon4 = "pabellon4.csv", pabellon5 = "pabellon5.csv", cedula = null, numeroPPL = null

filas = 801, filas5 = 180, filasPabe234 = 181, filasPabe1 = 81, columnas = 13, opcion = 0, seguir = 0, p1 = 1, p2 = 2, p3 = 3, p4 = 4, p5 = 5

Arreglo - promedioPenas[10]

principal[filas][columnas]

mpabellon1[filasPabe1][columnas]

mpabellon2[filasPabe234][columnas]

mpabellon3[filasPabe234][columnas]

mpabellon4[filasPabe234][columnas]

mpabellon5[filas5][columnas]

Método principal

Hacer

Mostrar opcionesInicio()

Leer opcion

Seleccionar (opción)

caso 1:

Mostrar "ABRIENDO ARCHIVO..."

abrirArchivoInfo(archivoInfo)

caso 2:

Mostrar opcionesBaseDatos()

Leer opcion

seleccionar (opción)

caso 1:

Mostrar "ABRIENDO ARCHIVO..."

leercsv(principal, archivoPrincipal)

abrirArchivoCompleto(archivoPrincipal, principal, filas)

caso 2:

Mostrar opcionesDatosPabellones()

Leer opcion

seleccionar (opción)

caso 1:

Mostrar "ABRIENDO ARCHIVO..."

leercsv(principal, archivoPrincipal)

abrirArchivoPabellon1(pabellon1, principal, filas)

caso 2:

Mostrar "ABRIENDO ARCHIVO..."

leercsv(principal, archivoPrincipal)

abrirArchivoPabellon2(pabellon2, principal, filas)

caso 3:

Mostrar "ABRIENDO ARCHIVO..."

leercsv(principal, archivoPrincipal)

abrirArchivoPabellon3(pabellon3, principal, filas)

caso 4:

Mostrar "ABRIENDO ARCHIVO..."

leercsv(principal, archivoPrincipal)

abrirArchivoPabellon4(pabellon4, principal, filas)

caso 5:

Mostrar "ABRIENDO ARCHIVO..."

leercsv(principal, archivoPrincipal)

abrirArchivoPabellon5(pabellon5, principal, filas)

caso contrario:

Mostrar "Opcion incorrecta"

caso 3:

Mostrar opcionesbuscar()

Leer opcion

seleccionar (opción)

caso 1:

Mostrar "Ingresa el número de identificación del PPL"

Leer cedula

buscarPPLcedula(cedula, principal, filas, archivoPrincipal)

caso 2:

Mostrar "Ingresa el número del PPL(1-800)"

Leer numeroPPL

buscarPPLnumero(numeroPPL, principal, filas, archivoPrincipal)

caso contrario:

Mostrar "Opcion incorrecta"

caso 4:

Mostrar "ESTADISTICAS DE LA CARCEL:"

leercsv(principal, archivoPrincipal)

leercsvPabe(mpabellon1, pabellon1)

leercsvPabe(mpabellon2, pabellon2)

leercsvPabe(mpabellon3, pabellon3)

leercsvPabe(mpabellon4, pabellon4)

leercsvPabe(mpabellon5, pabellon5)

promedioDePenas(filas, principal, archivoPrincipal, promedioPenas)
delitoMasCometido(promedioPenas)
promedioEdadesCompleto(principal, filas)
promedioEdadesPabe1(mpabellon1, p1)
promedioEdadesPabe1(mpabellon2, p2)
promedioEdadesPabe1(mpabellon3, p3)
promedioEdadesPabe1(mpabellon4, p4)
promedioEdadesPabe1(mpabellon5, p5)
modaCarcelEdadesPrincipal(principal)
modaCarcelEdades(mpabellon1, p1)
modaCarcelEdades(mpabellon2, p2)
modaCarcelEdades(mpabellon3, p3)
modaCarcelEdades(mpabellon4, p4)
modaCarcelEdades(mpabellon5, p5)

caso 5:

Hacer

Mostrar "Ingrese la clave de acceso (12345)"

Leer contraseña

Si contraseña IgualA "12345" entonces

Mostrar "ACCESO EXITOSO"

Mostrar opcionesAdministracion()

Leer opcion

seleccionar (opción)

caso 1:

Mostrar consultarCuposDisponibles(principal, filas, archivoPrincipal)

caso 2:

Mostrar opcionesCastigos()

Leer opción

seleccionar (opción)

caso 1:

Mostrar "Ingresa la identificación del PPL"

Leer cedula

disminuirvisita(cedula, principal, filas, archivoPrincipal, pabellon1, pabellon2, pabellon3, pabellon4, pabellon5)

caso 2:

Mostrar "Ingresa la identificacion del PPL"

Leer cedula

aumentarpena(cedula, principal, filas, archivoPrincipal, pabellon1, pabellon2, pabellon3, pabellon4, pabellon5)

caso contrario:

Mostrar "Opcion Incorrecta"

caso 3:

Mostrar "Ingresa la identificacion del PPL a trasladar"

Leer cedula

trasladarPPL(cedula, principal, filas, archivoPrincipal, pabellon1, pabellon2, pabellon3, pabellon4, pabellon5)

caso 4:

ingresarPPL(principal, filas, archivoPrincipal, cedula, pabellon1, pabellon2, pabellon3, pabellon4, pabellon5)

caso contrario:

Mostrar "Opcion incorrecta"

sino

Mostrar "Clave de acceso incorrecta"

mientras contraseña no sea igual a "12345"

caso contrario:

Mostrar "Opcion incorrecta"

Mostrar "DESEA CONTINUAR CON EL PROGRAMA?([1]SI/[2]NO)"

Leer seguir

Si seguir IgualA 2 entonces

Mostrar "GRACIAS POR USAR ESTE PROGRAMA"

Mientras seguir sea igual a 1

Función opcionesInicio() -> Cadena

cadena = "SISTEMA DE CONTROL CARCELARIA",

"Ingresa el numero correspondiente a la opcion deseada",

"[1] -> Ver información de la cárcel",

"[2] -> Ingresar a la base de datos",

```
"[3] -> Buscar PPL especifico",  
"[4] -> Observar estadísticas",  
"[5] -> Administración (SOLO PERSONAL AUTORIZADO)"
```

Retornar cadena

FinFunción

Función opcionesBaseDatos() -> Cadena

```
cadena = "Ingrese el numero correspondiente a la opcion deseada",  
"[1] -> Ver la base de datos completa",  
"[2] -> Ver la base de datos distribuida por pabellones"
```

Retornar cadena

FinFunción

Función opcionesDatosPabellones() -> Cadena

```
cadena = "Ingrese el numero correspondiente al pabellon que desea verificar",  
"[1] -> Pabellon 1",  
"[2] -> Pabellon 2",  
"[3] -> Pabellon 3",  
"[4] -> Pabellon 4",  
"[5] -> Pabellon 5"
```

Retornar cadena

FinFunción

Función opcionesBuscar() -> Cadena

```
cadena = "Como deseas buscar al PPL",  
"[1] -> Identificacion",  
"[2] -> Numero"
```

Retornar cadena

FinFunción

Función opcionesAdministracion() -> Cadena

```
cadena = "Ingrese el numero correspondiente a la opcion deseada",  
"[1] -> Consultar cupos disponibles",  
"[2] -> Castigar PPLs",  
"[3] -> Trasladar PPL",  
"[4] -> Ingresar PPL"
```

Retornar cadena

FinFunción

Función opcionesCastigos() -> Cadena

```
cadena = "Ingrese el numero correspondiente a la opcion deseada",  
        "[1] -> Disminuir horas de visita",  
        "[2] -> Aumentar pena"
```

Retornar cadena

FinFunción

Función opcionesDisminuirVisita() -> Cadena

```
cadena = "Cuántas horas deseas disminuir?",  
        "[1] -> 1 hora",  
        "[2] -> 2 horas",  
        "[3] -> 3 horas",  
        "[4] -> 4 horas",  
        "NOTA: Si no se puede restar más, se queda en 0"
```

Retornar cadena

FinFunción

Procedimiento leerCSV(principal: s , archivoPrincipal: s) : s

```
linea = Nulo
```

```
i = 0
```

```
arreglo[]
```

```
leer = Escanear(archivoPrincipal)
```

Mientras leer Hacer

```
linea = leer
```

```
arreglo[] = linea
```

```
principal[i][0] = arreglo[0]
```

```
principal[i][1] = arreglo[1]
```

```
principal[i][2] = arreglo[2]
```

```
principal[i][3] = arreglo[3]
```

```
principal[i][4] = arreglo[4]
```

```
principal[i][5] = arreglo[5]
```

```
principal[i][6] = arreglo[6]
```

```
principal[i][7] = arreglo[7]
principal[i][8] = arreglo[8]
principal[i][9] = arreglo[9]
principal[i][10] = arreglo[10]
principal[i][11] = arreglo[11]
principal[i][12] = arreglo[12]
i = i + 1
```

FinMientras

FinProcedimiento

Procedimiento leerCSVPabellon(mpabellon: s, pabellon: s) : s

```
i = 0
leer = Escanear(pabellon)
Mientras leer Hacer
    Arreglo[] = leer
    mpabellon[i][0] = arreglo[0]
    mpabellon[i][1] = arreglo[1]
    mpabellon[i][2] = arreglo[2]
    mpabellon[i][3] = arreglo[3]
    mpabellon[i][4] = arreglo[4]
    mpabellon[i][5] = arreglo[5]
    mpabellon[i][6] = arreglo[6]
    mpabellon[i][7] = arreglo[7]
    mpabellon[i][8] = arreglo[8]
    mpabellon[i][9] = arreglo[9]
    mpabellon[i][10] = arreglo[10]
    mpabellon[i][11] = arreglo[11]
    mpabellon[i][12] = arreglo[12]
    i = i + 1
```

FinMientras

FinProcedimiento

// Estadísticas

Procedimiento promedioDePenas(filas: i, principal: s, archivoPrincipal: s, promedioPenas: s) : s

```
lector = Escanear(archivoPrincipal)
linea = Nulo
Mientras (linea = lector NoSeaNulo Hacer
    Si (linea = ("Narcotrafico"))
        promedioPenas[0] = promedioPenas[0] + 1
    Si (línea = ("Homicidio"))
        promedioPenas[1] = promedioPenas[1] + 1
    Si (línea = ("Robo"))
        promedioPenas[2] = promedioPenas[2] + 1
    Si (línea = ("Hurto"))
        promedioPenas[3] = promedioPenas[3] + 1
    Si (línea = ("Fraude"))
        promedioPenas[4] = promedioPenas[4] + 1
    Si (línea = ("Evasion"))
        promedioPenas[5] = promedioPenas[5] + 1
    Si (línea = ("Invasion"))
        promedioPenas[6] = promedioPenas[6] + 1
    Si (línea = ("Amenazas"))
        promedioPenas[7] = promedioPenas[7] + 1
    Si (línea = ("Estafa"))
        promedioPenas[8] = promedioPenas[8] + 1
    Si (línea = ("Otros"))
        promedioPenas[9] = promedioPenas[9] + 1
FinMientras
Mostrar "Cantidad de PPL por Delito"
Mostrar "(Alto)PPL Narcotrafico: ", promedioPenas[0]
Mostrar "(Alto)PPL Homicidio: ", promedioPenas[1]
Mostrar "(Medio)PPL Robo: ", promedioPenas[2]
Mostrar "(Medio)PPL Hurto: ", promedioPenas[3]
Mostrar "(Medio)PPL Fraude: ", promedioPenas[4]
Mostrar "(Medio)PPL Evasion: ", promedioPenas[5]
Mostrar "(Bajo)PPL Invasion: ", promedioPenas[6]
```

Mostrar "(Bajo)PPL Amenazas: ", promedioPenas[7]

Mostrar "(Bajo)PPL Estafa: ", promedioPenas[8]

Mostrar "(Bajo)PPL otros: ", promedioPenas[9]

FinProcedimiento

Procedimiento promedioEdadesCompleto(principal: s, filas: i) : s

promedioEdades = 0

Para i = 1 Hasta filas Hacer

promedioEdades = promedioEdades + principal[i][3]

FinPara

promedioEdades = promedioEdades / (filas-1)

Mostrar "Promedio Edad (carcel): ", promedioEdades, " años"

FinProcedimiento

Procedimiento promedioEdadesPabe1(mpabellon: s, p: i) : s

promedioEdades = 0

cont = 0

Para i = 1 Hasta Longitud(mpabellon) Hacer

Si mpabellon[i][3] NoEsNulo Entonces

promedioEdades = promedioEdades + mpabellon[i][3]

FinSi

Cont = cont + 1

FinPara

promedioEdades = promedioEdades / cont

Mostrar "Promedio de edad del Pabellon ", p, ":", promedioEdades, " años"

FinProcedimiento

Procedimiento delitoMasCometido(promedioPenas: i) : s

// Alto

Si promedioPenas[0] > promedioPenas[1] Entonces

Mostrar "Delito más cometido(Alto): Narcotrafico"

Sino

Mostrar "Delito más cometido(Alto): Homicidio"

FinSi

// Medio

Si promedioPenas[2] > promedioPenas[3] Y promedioPenas[2] > promedioPenas[4]
Y promedioPenas[2] > promedioPenas[5] Entonces

Mostrar "Delito mas cometido(Medio): Robo"

Si promedioPenas[3] > promedioPenas[2] Y promedioPenas[3] > promedioPenas[4]
Y promedioPenas[3] > promedioPenas[5] Entonces

Mostrar "Delito mas cometido(Medio): Hurto"

Si promedioPenas[4] > promedioPenas[3] Y promedioPenas[4] > promedioPenas[2]
Y promedioPenas[4] > promedioPenas[5] Entonces

Mostrar "Delito mas cometido(Medio): Fraude"

Si promedioPenas[5] > promedioPenas[3] Y promedioPenas[5] > promedioPenas[4]
Y promedioPenas[5] > promedioPenas[2] Entonces

Mostrar "Delito mas cometido(Medio): Evasion"

FinSi

// Bajo

Si promedioPenas[6] > promedioPenas[7] Y promedioPenas[6] > promedioPenas[8]
Y promedioPenas[6] > promedioPenas[9] Entonces

Mostrar "Delito mas cometido(Bajo): Invasion"

Si promedioPenas[7] > promedioPenas[6] Y promedioPenas[7] > promedioPenas[8]
Y promedioPenas[7] > promedioPenas[9] Entonces

Mostrar "Delito mas cometido(Bajo): Amenazas"

Si promedioPenas[8] > promedioPenas[6] Y promedioPenas[8] > promedioPenas[7]
Y promedioPenas[8] > promedioPenas[9] Entonces

Mostrar "Delito mas cometido(Bajo): Estafa"

Si promedioPenas[9] > promedioPenas[6] Y promedioPenas[9] > promedioPenas[7]
Y promedioPenas[9] > promedioPenas[8] Entonces

Mostrar "Delito mas cometido(Bajo): Otros"

FinSi

// Calcular la moda de edades en la cárcel principal

Procedimiento modaCarcelEdadesPrincipal(matriz: s) : s

maximoNumRepeticiones = 0

moda = 0

Para i = 1 Hasta Longitud(matriz) Hacer

numRepeticiones = 0

Para j = 1; Hasta Longitud(matriz); Hacer

Si matriz[i][3] IgualA matriz[j][3] Entonces

```

        numRepeticiones = numRepeticiones + 1
    FinSi
FinPara
Si numRepeticiones > maximoNumRepeticiones Entonces
    moda = matriz[i][3]
    maximoNumRepeticiones = numRepeticiones
FinSi
FinPara
Mostrar "Edad mas repetida (carcel): ", moda
FinProcedimiento

// Calcular la moda de edades en un pabellón específico
Procedimiento modaCarcelEdades(matriz: s, p: d) : s
    maximoNumRepeticiones = 0
    moda = 0
    Para i = 1 Hasta Longitud(matriz) Hacer
        numRepeticiones = 0
        Para j = 1 Hasta Longitud(matriz) Hacer
            Si matriz[i][3] IgualA matriz[j][3] Entonces
                numRepeticiones = numRepeticiones + 1
            FinSi
        FinPara
        Si numRepeticiones > maximoNumRepeticiones Entonces
            moda = matriz[i][3]
            maximoNumRepeticiones = numRepeticiones
        FinSi
    FinPara
    Mostrar "Edad mas repetida Pabellon ", p, ": ", moda
FinProcedimiento

// Métodos para abrir archivos específicos
Procedimiento abrirArchivoInfo(archivoInfo: s) : s
    // Ruta
    rutaArchivo = ObtenerRutaActual(), archivoInfo

```


abrirArchivos(rutaArchivo)

Mostrar "Puedes visualizar la informacion en el archivo ", archivoInfo

FinProcedimiento

Procedimiento abrirArchivoCompleto(archivoPrincipal: s, principal: s, filas: i) : s

actualizarArchivoPrincipal(archivoPrincipal, principal, filas)

// Ruta

rutaArchivo = ObtenerRutaActual(), archivoPrincipal

abrirArchivos(rutaArchivo)

Mostrar "Puedes visualizar la informacion en el archivo ", archivoPrincipal

FinProcedimiento

Procedimiento abrirArchivoPabellon1(pabellon1: s, principal: s, filas: i) : s

actualizarPabellon1(pabellon1, principal, filas)

// Ruta

rutaArchivo = ObtenerRutaActual(), pabellon1

abrirArchivos(rutaArchivo)

Mostrar "Puedes visualizar la informacion en el archivo ", pabellon1

FinProcedimiento

Procedimiento abrirArchivoPabellon2(pabellon2: s, principal: s, filas: i) : s

actualizarPabellon2(pabellon2, principal, filas)

// Ruta

rutaArchivo = ObtenerRutaActual(), pabellon2

abrirArchivos(rutaArchivo)

Mostrar "Puedes visualizar la informacion en el archivo ", pabellon2

FinProcedimiento

Procedimiento abrirArchivoPabellon3(pabellon3: s, principal: s, filas: i) : s

actualizarPabellon3(pabellon3, principal, filas)

// Ruta

rutaArchivo = ObtenerRutaActual(), pabellon3

abrirArchivos(rutaArchivo)

Mostrar "Puedes visualizar la informacion en el archivo ", pabellon3

FinProcedimiento

Procedimiento abrirArchivoPabellon4(pabellon4: s, principal: s, filas: i) : s

```

    actualizarPabellon4(pabellon4, principal, filas)
// Ruta
    rutaArchivo = ObtenerRutaActual(), pabellon4
        abrirArchivos(rutaArchivo)
            Mostrar "Puedes visualizar la informacion en el archivo ", pabellon4
FinProcedimiento

Procedimiento abrirArchivoPabellon5(pabellon5: s, principal: s, filas: i) : s
    actualizarPabellon5(pabellon5, principal, filas)
// Ruta
    rutaArchivo = ObtenerRutaActual(), pabellon5
        abrirArchivos(rutaArchivo)
            Mostrar "Puedes visualizar la informacion en el archivo ", pabellon5
FinProcedimiento

// Métodos para buscar PPLs

Procedimiento buscarPPLcedula(cedula: s, principal: s, filas: i, archivoPrincipal: s) : s
    leercsv(principal, archivoPrincipal)
    Para i = 1 Hasta filas Hacer
        Si cedula IgualA principal[i][1] Entonces

Mostrar"#PPL|Cedula|Nombres|Edad(Anios)|Pena(anios)|Fecha(Ingreso)(d/m/a)|Fecha(Sal
ida)(d/m/a)|AniosRestantes(pena)|Pabellon|Celda|Clasificacion|Visitas
semanales(Horas)|Delito"

Mostrar principal[i][0], principal[i][1], principal[i][2], principal[i][3], principal[i][4],
principal[i][5], principal[i][6], principal[i][7], principal[i][8], principal[i][9], principal[i][10],
principal[i][11], principal[i][12]

        Salir
    Sino Si cedula IgualA principal[i][1] Y (i IgualA filas - 1) Entonces
        Mostrar ("PPL NO ENCONTRADO")
    FinSi
FinPara
FinProcedimiento

Procedimiento buscarPPLnumero(numeroPPL: s, principal: s, filas: i, archivoPrincipal: s) : s
    leercsv(principal, archivoPrincipal)
    numeroPPL = "PPL ", numeroPPL

```

```

Para i = 1 Hasta filas Hacer
    Si numeroPPL IgualA principal[i][0] Entonces
        Si principal[i][1] IgualA "libre" Entonces
            Mostrar "Este PPL fue trasladado, espacio disponible"
        Sino
            Mostrar"#PPL|Cedula|Nombres|Edad(Anios)|Pena(anios)|Fecha(Ingreso)(d/m/a)|Fecha(Salida)(d/m/a)|AniosRestantes(pena)|Pabellon|Celda|Clasificacion|Visitas semanales(Horas)|Delito"
            Mostrar principal[i][0], principal[i][1], principal[i][2], principal[i][3], principal[i][4], principal[i][5], principal[i][6], principal[i][7], principal[i][8], principal[i][9], principal[i][10], principal[i][11], principal[i][12]
        FinSi
    Salir
    Sino Si i IgualA (filas - 1) Entonces
        Mostrar "PPL NO ENCONTRADO"
    FinSi
FinPara
FinProcedimiento
// Administración
Procedimiento trasladarPPL(cedula: s, principal: s, filas: i, archivoPrincipal: s, pabellon1: s, pabellon2: s, pabellon3: s, pabellon4: s, pabellon5: s) : s
    leercsv(principal, archivoPrincipal)
    Para i = 1 Hasta filas Hacer
        Si cedula IgualA principal[i][1] Entonces
            principal[i][1] = "libre"
            principal[i][2] = "libre"
            principal[i][3] = "libre"
            principal[i][4] = "libre"
            principal[i][5] = "libre"
            principal[i][6] = "libre"
            principal[i][7] = "libre"
            principal[i][11] = "libre"
            principal[i][12] = "libre"
            actualizarArchivoPrincipal(archivoPrincipal, principal, filas)
            actualizarPabellon1(pabellon1, principal, filas)

```

```

    actualizarPabellon2(pabellon2, principal, filas)
    actualizarPabellon3(pabellon3, principal, filas)
    actualizarPabellon4(pabellon4, principal, filas)
    actualizarPabellon5(pabellon5, principal, filas)
    Mostrar "PPL trasladado con exito"
    Salir
Sino Si i IgualA (filas - 1) Entonces
    Mostrar "PPL NO ENCONTRADO"
FinSi
FinPara
FinProcedimiento
Función consultarCuposDisponibles(principal, filas, archivoPrincipal)
    leercsv(principal, archivoPrincipal)
    cadena = "Cupos Disponibles"
    hayCupos = Falso
    Para i = 1 Hasta filas Hacer
        Si principal[i][1] IgualA "libre" Entonces
            hayCupos = Verdadero
            cadena = "Cupo-> ", principal[i][0], " - Celda: ", principal[i][9], " - Pabellon: ",
                principal[i][8]
        FinSi
    FinPara
    Si hayCupos Entonces
        Devolver cadena
    Sino
        Devolver "No hay cupos disponibles"
    FinSi
FinFunción
Procedimiento disminuirvisita(cedula: s, principal: s, filas: i, archivoPrincipal: s, pabellon1:
s, pabellon2: s, pabellon3: s, pabellon4: s, pabellon5: s) : s
    disminuir = 0
    visitaEntero = 0
    leercsv(principal, archivoPrincipal)

```

Para i = 1 Hasta filas Hacer

Si cedula IgualA principal[i][1] Entonces

Mostrar "Este PPL tiene: ", principal[i][11], " horas de visita a la semana"

Mostrar opcionesDisminuirVisita()

Leer disminuir

Según disminuir Hacer

Caso 1:

visitaEntero = principal[i][11]

visitaEntero = visitaEntero - 1

Si visitaEntero < 0 Entonces

visitaEntero = 0

FinSi

principal[i][11] = visitaEntero

Caso 2:

visitaEntero = principal[i][11]

visitaEntero = visitaEntero - 2

Si visitaEntero < 0 Entonces

visitaEntero = 0

FinSi

principal[i][11] = visitaEntero

Caso 3:

visitaEntero = principal[i][11]

visitaEntero = visitaEntero - 3

Si visitaEntero < 0 Entonces

visitaEntero = 0

FinSi

principal[i][11] = visitaEntero

Caso 4:

visitaEntero = principal[i][11]

visitaEntero = visitaEntero - 4

Si visitaEntero < 0 Entonces

visitaEntero = 0

FinSi

principal[i][11] = visitaEntero

De Otro Modo:

Mostrar "Opción incorrecta"

FinSegún

actualizarArchivoPrincipal(archivoPrincipal, principal, filas)

actualizarPabellon1(pabellon1, principal, filas)

actualizarPabellon2(pabellon2, principal, filas)

actualizarPabellon3(pabellon3, principal, filas)

actualizarPabellon4(pabellon4, principal, filas)

actualizarPabellon5(pabellon5, principal, filas)

Mostrar "Horas de visita actualizadas con éxito"

Mostrar "Puedes consultar los cambios en el buscador (opcion[3])"

Salir

Sino Si i IgualA (filas - 1) Entonces

Mostrar "PPL NO ENCONTRADO"

FinSi

FinPara

FinProcedimiento

Procedimiento aumentarPena(cedula: s, principal: s, filas: i, archivoPrincipal: s, pabellon1: s, pabellon2: s, pabellon3: s, pabellon4: s, pabellon5: s) : s

leercsv(principal, archivoPrincipal)

aumento = 0

Para i = 1 Hasta filas Hacer

Si cedula IgualA principal[i][1] Entonces

Mostrar "Este PPL tiene: ", principal[i][4], " años de pena"

Mostrar "Le quedan: ", principal[i][7], " años por cumplir"

Mostrar "Ingresa el numero de años que deseas aumentar"

Mostrar "NOTA: Solo se pueden aumentar hasta 10 años"

Leer aumento

Mientras aumento <= 0 O aumento > 10 Hacer

Mostrar "El aumento puede ser: máximo 10 años, mínimo 1 año"

Mostrar "Ingresa nuevamente la cantidad"

Leer aumento

FinMientras

aumento = aumento + principal[i][4]

principal[i][4] = aumento

actualizarArchivoPrincipal(archivoPrincipal, principal, filas)

actualizarPabellon1(pabellon1, principal, filas)

actualizarPabellon2(pabellon2, principal, filas)

actualizarPabellon3(pabellon3, principal, filas)

actualizarPabellon4(pabellon4, principal, filas)

actualizarPabellon5(pabellon5, principal, filas)

Mostrar "Pena actualizada con éxito"

Mostrar "Puedes consultar los cambios en el buscador (opcion[3])"

Salir

Sino Si i IgualA (filas - 1) Entonces

Mostrar "PPL NO ENCONTRADO"

FinSi

FinPara

FinProcedimiento

Procedimiento ingresarPPL(principal: s, filas: i, archivoPrincipal: s, cedula: s, pabellon1: s, pabellon2: s, pabellon3: s, pabellon4: s, pabellon5: s) : s

cupo = 0

Para i = 1 Hasta filas Hacer

Si consultarCuposDisponibles(principal, filas, archivoPrincipal) IgualA "No hay cupos disponibles" Entonces

Mostrar "No hay cupos disponibles en este momento"

Salir

FinSi

Mostrar consultarCuposDisponibles(principal, filas, archivoPrincipal)

Mostrar "Ingresa el numero del PPL correspondiente al cupo que deseas ocupar"

Mostrar "NOTA: Debes ingresar al PPL a un pabellón según sus delitos"

Leer cupo

numeroPPL = "PPL ", cupo

```
Si numeroPPL igualA principal[cupo][0] Entonces
    Mostrar "Ingresa la identificación del PPL"
    Leer principal[cupo][1]
    Mostrar "Ingresa el primer nombre"
    Leer nombre1
    Mostrar "Ingresa el segundo nombre"
    Leer nombre2
    Mostrar "Ingresa el primer apellido"
    Leer apellido1
    Mostrar "Ingresa el segundo apellido"
    Leer apellido2
    principal[cupo][2] = nombre1, nombre2, apellido1, apellido2
    Mostrar "Ingresa la edad del PPL"
    Leer principal[cupo][3]
    Mostrar "Ingresa la pena a cumplir"
    Leer principal[cupo][4]
    Mostrar "Ingresa las visitas semanales"
    Leer principal[cupo][11]
    Mostrar "Ingresa el delito cometido"
    Leer principal[cupo][12]
    fechaActual = FechaActual()
    diaActual = (fechaActual)
    mesActual = (fechaActual)
    anioActual = (fechaActual)
    principal[cupo][5] = diaActual, (mesActual), anioActual
    aniosalida = anioActual + principal[cupo][4]
    principal[cupo][6] = diaActual, (mesActual), aniosalida
    aniosrestantes = aniosalida - anioActual
    principal[cupo][7] = aniosrestantes
    actualizarArchivoPrincipal(archivoPrincipal, principal, filas)
    actualizarPabellon1(pabellon1, principal, filas)
    actualizarPabellon2(pabellon2, principal, filas)
```


actualizarPabellon3(pabellon3, principal, filas)

actualizarPabellon4(pabellon4, principal, filas)

actualizarPabellon5(pabellon5, principal, filas)

Mostrar "PPL ingresado con éxito"

Mostrar"#PPL|Cedula|Nombres|Edad(Anios)|Pena(anios)|Fecha(Ingreso)(d/m/a)|Fecha(Salida)(d/m/a)|AniosRestantes(pena)|Pabellon|Celda|Clasificacion|Visitas semanales(Horas)|Delito"

Mostrar principal[cupo][0], principal[cupo][1], principal[cupo][2], principal[cupo][3], principal[cupo][4], principal[cupo][5], principal[cupo][6], principal[cupo][7], principal[cupo][8], principal[cupo][9], principal[cupo][10], principal[cupo][11], principal[cupo][12]

Salir

Sino Si ((numeroPPL IgualA principal[i][0]) Y (i IgualA (filas - 1)) Entonces

Mostrar "Numero incorrecto"

FinSi

FinPara

FinProcedimiento

Procedimiento actualizarArchivoPrincipal(archivoPrincipal: s, principal: s, filas: d) : s

e = escribir(principal)

e="#PPL;Cedula;Nombres;Edad(Anios);Pena(anios);Fecha(Ingreso)(d/m/a);Fecha(Salida)(d/m/a);Anios Restantes (pena);Pabellon;Celda;Clasificacion;Visitas semanales(Horas);Delito"

Para i = 1 Hasta filas Hacer

e = principal[i][0], principal[i][1], principal[i][2], principal[i][3], principal[i][4], principal[i][5], principal[i][6], principal[i][7], principal[i][8], principal[i][9], principal[i][10], principal[i][11], principal[i][12]

FinPara

FinProcedimiento

Procedimiento actualizarPabellon1(pabellon1: s, principal: s, filas: d) : s

e = escribir(pabellon1)

e="#PPL;Cedula;Nombres;Edad(Anios);Pena(anios);Fecha(Ingreso)(d/m/a);Fecha(Salida)(d/m/a);Anios Restantes (pena);Pabellon;Celda;Clasificacion;Visitas semanales(Horas);Delito"

Para i = 1 Hasta 80 Hacer

e = principal[i][0], principal[i][1], principal[i][2], principal[i][3], principal[i][4], principal[i][5], principal[i][6], principal[i][7], principal[i][8], principal[i][9], principal[i][10], principal[i][11], principal[i][12]

FinPara

FinProcedimiento

Procedimiento actualizarPabellon2(pabellon2: s, principal: s, filas: d) : s

e = escribir(pabellon2)

e="#PPL;Cedula;Nombres;Edad(Anios);Pena(anios);Fecha(Ingreso)(d/m/a);Fecha(Salida)(d/m/a);Anios Restantes (pena);Pabellon;Celda;Clasificacion;Visitas semanales(Horas);Delito"

Para i = 81 Hasta 260 Hacer

e = principal[i][0], principal[i][1], principal[i][2], principal[i][3], principal[i][4], principal[i][5], principal[i][6], principal[i][7], principal[i][8], principal[i][9], principal[i][10], principal[i][11], principal[i][12]

FinPara

FinProcedimiento

Procedimiento actualizarPabellon3(pabellon3: s, principal: s, filas: d) : s

e = escribir(pabellon3)

e="#PPL;Cedula;Nombres;Edad(Anios);Pena(anios);Fecha(Ingreso)(d/m/a);Fecha(Salida)(d/m/a);Anios Restantes (pena);Pabellon;Celda;Clasificacion;Visitas semanales(Horas);Delito"

Para i = 261 Hasta 440 Hacer

e = principal[i][0], principal[i][1], principal[i][2], principal[i][3], principal[i][4], principal[i][5], principal[i][6], principal[i][7], principal[i][8], principal[i][9], principal[i][10], principal[i][11], principal[i][12]

FinPara

FinProcedimiento

Procedimiento actualizarPabellon4(pabellon4: s, principal: s, filas: d) : s

e = escribir(pabellon4)

e="#PPL;Cedula;Nombres;Edad(Anios);Pena(anios);Fecha(Ingreso)(d/m/a);Fecha(Salida)(d/m/a);Anios Restantes (pena);Pabellon;Celda;Clasificacion;Visitas semanales(Horas);Delito"

Para i = 441 Hasta 620 Hacer

e = principal[i][0], principal[i][1], principal[i][2], principal[i][3], principal[i][4], principal[i][5], principal[i][6], principal[i][7], principal[i][8], principal[i][9], principal[i][10], principal[i][11], principal[i][12]

FinPara

FinProcedimiento

Procedimiento actualizarPabellon5(pabellon5: s, principal: s, filas: d) : s

intentar

e = escribir(pabellon5)

e="#PPL;Cedula;Nombres;Edad(Anios);Pena(anios);Fecha(Ingreso)(d/m/a);Fecha(Salida)(d/m/a);Anios Restantes (pena);Pabellon;Celda;Clasificacion;Visitas semanales(Horas);Delito"

Para i = 621 Hasta 799 Hacer

e = principal[i][0], principal[i][1], principal[i][2], principal[i][3], principal[i][4], principal[i][5], principal[i][6], principal[i][7], principal[i][8], principal[i][9], principal[i][10], principal[i][11], principal[i][12]

FinPara

FinProcedimiento

Fin

PRUEBA DE ESCRITORIO

SISTEMA DE CONTROL CARCELARIA

Ingrese el numero correspondiente a la opción deseada

[1] -> Ver información de la cárcel

[2] -> Ingresar a la base de datos

[3] -> Buscar PPL especifico

[4] -> Observar estadísticas

[5] -> Administracion (SOLO PERSONAL AUTORIZADO)

1

ABRIENDO ARCHIVO...

Puedes visualizar la informacion en el archivo informacionCarcel.txt

DESEA CONTINUAR CON EL PROGRAMA?([1]SI/[2]NO)

SISTEMA DE CONTROL CARCELARIA

Ingrese el numero correspondiente a la opcion deseada

[1] -> Ver informacion de la carcel

[2] -> Ingresar a la base de datos

[3] -> Buscar PPL especifico

[4] -> Observar estadísticas

[5] -> Administracion (SOLO PERSONAL AUTORIZADO)

4

ESTADISTICAS DE LA CARCEL:

Cantidad de PPL por Delito

(Alto)PPL Narcotrafico: 37

(Alto)PPL Homicidio: 43

(Medio)PPL Robo: 84

(Medio)PPL Hurto: 91

(Medio)PPL Fraude: 90

(Medio)PPL Evasion: 95

(Bajo)PPL Invasion: 88

(Bajo)PPL Amenazas: 81

(Bajo)PPL Estafa: 103

(Bajo)PPL otros: 88

Delito mas cometido(Alto): Homicidio

Delito mas cometido(Medio): Robo

Delito mas cometido(Bajo): Estafa

Promedio Edad (carcel): 45.42 anios

Promedio de edad del Pabellon 1: 45.05 anios

Promedio de edad del Pabellon 2: 45.42 anios

Promedio de edad del Pabellon 3: 45.29 anios

Promedio de edad del Pabellon 4: 46.37 anios

Promedio de edad del Pabellon 5: 44.75 anios

Edad mas repetida (carcel): 46

Edad mas repetida Pabellon 1: 48

Edad mas repetida Pabellon 2: 45

Edad mas repetida Pabellon 3: 46

Edad mas repetida Pabellon 4: 49

Edad mas repetida Pabellon 5: 44

DESEA CONTINUAR CON EL PROGRAMA?([1]SI/[2]NO)

2

GRACIAS POR USAR ESTE PROGRAMA