

Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave 36276

Alumno: Solano Meza Angel Daniel

Matrícula: 372453

Maestro: Pedro Núñez Yépiz

Actividad No. : 13

Tema-Unidad : Archivos Binarios

Ensenada Baja California a 12 de Noviembre del 2023

```

void menu()
{
    srand(time(NULL));
    TReg VectReg[5000], temp;
    int op, j, i, apagar, buscar, encontrado, ordenado, cargado, confirmar, creado, registros, editar, campo, sexo, puesto;
    char NombreTXT[50], Mostrar[50], NombreBIN[11] = {"datos.dll"};
    char Nombre[60];
    char puestos[6][11] = {"GERENTE", "TECNICO", "EMPLEADO", "ASISTENTE", "TECNICO", "ANALISTA"};
    char sexos[2][7] = {"HOMBRE", "MUJER"};
    i = 0;
    ordenado = 0;
    cargado = 0;
    creado = 0;
    do
    {
        op = msges();
        switch (op)
        {
            case 1: // AGREGAR
                system("CLS");
                for (j = 0; j < AUTO; j++)
                {
                    temp = RegistroAuto();
                    while (BusquedaTReg(VectReg, i, temp.key) != -1)
                    {
                        temp.key = NumAleatorio(1, 2000);
                    }
                    if (i <= N)
                    {
                        VectReg[i++] = temp;
                        ordenado = 0;
                    }
                }
                if (i < N)
                {
                    printf("REGISTROS LLENADOS AUTOMATICAMENTE\n");
                }
            }
        }
    } while (1);
}

```

```

    }
    else
    {
        printf("500 REGISTROS OCUPADOS\n");
    }
    system("PAUSE");
    break;
case 2: // EDITAR REGISTRO
    system("CLS");
    if (i == 0)
    {
        printf("REGISTROS VACIOS\n");
        system("PAUSE");
    }
    else
    {
        printf("NUMERO DE EMPLEADO A BUSCAR\n");
        buscar = Validar(1, 2000);
        if (ordenado == 0)
        {
            encontrado = BusquedaTReg(VectReg, i, buscar);
        }
        else
        {
            encontrado = BusquedaBinaria(VectReg, 0, i, buscar);
        }
        if (encontrado == -1)
        {
            printf("NUMERO DE EMPLEADO NO ENCONTRADA\n");
        }
        else
        {
            if (VectReg[encontrado].Status == 0)
            {
                printf("NUMERO DE EMPLEADO DE ALUMNO DESACTIVADO\n");
            }
            else
            {

```

```

system("CLS");
printf("DATOS DEL REGISTRO ENCONTRADO:\n");
printf("EMPLEADO EN REGISTRO: %d\n", encontrado);
printf("NUMERO DE EMPLEADO: %d\n", VectReg[encontrado].Empleado);
printf("NOMBRE: %s\n", VectReg[encontrado].Nombre);
printf("APELLIDO PATERNO: %s\n", VectReg[encontrado].Paterno);
printf("APELLIDO MATERNO: %s\n", VectReg[encontrado].Materno);
printf("PUESTO: %s\n", VectReg[encontrado].Puesto);
printf("EDAD: %d\n", VectReg[encontrado].Edad);
printf("SEXO: %s\n", VectReg[encontrado].Sexo);
system("pause");
system("cls");
printf("EDITAR REGISTRO: 1.SI 0.NO\n");
editar = Validar(0, 1);
system("cls");
if (editar)
{
    printf("---EDITAR REGISTRO---\n");
    printf("1. NUMERO DE EMPLEADO\n");
    printf("2. NOMBRE DE EMPLEADO\n");
    printf("3. PUESTO\n");
    printf("4. SEXO\n");
    printf("5. EDAD\n");
    printf("REGISTRO A VALIDAR: ");
    campo = Validar(1, 5);
    system("cls");
    switch (campo)
    {
        case 1:
            do
            {
                printf("EDITAR NUMERO DE EMPLEADO\n");
                temp.Empleado = Validar(1, 2000);
                temp.key = temp.Empleado;
            } while (BusquedaTReg(VectReg, i, temp.key) != -1);
            VectReg[encontrado].Empleado = temp.Empleado;
            VectReg[encontrado].key = temp.Empleado;

```

```

        break;
    case 2:
        ValiCadena(Nombre, "EDITAR NOMBRE DEL EMPLEADO");
        strcpy(VectReg[encontrado].Nombre, Nombre);
        ValiCadena(Nombre, "EDITAR APELLIDO PATERNO DEL EMPLEADO");
        strcpy(VectReg[encontrado].Paterno, Nombre);
        ValiCadena(Nombre, "EDITAR APELLIDO MATERNO DEL EMPLEADO");
        strcpy(VectReg[encontrado].Materno, Nombre);
        break;
    case 3:
        printf("EDITAR PUESTO\n");
        for (int i = 0; i < 5; i++)
        {
            printf("%d.- %s\n", i, puestos[i]);
        }
        printf("SELECCIONA UN PUESTO \n");
        puesto = Validar(0, 4);
        strcpy(temp.Puesto, puestos[puesto]);
        strcpy(VectReg[encontrado].Puesto, temp.Puesto);
        break;
    case 4:
        printf("EDITAR SEXO\n");
        printf("1. MUJER\n0. HOMBRE\n");
        printf("SELECCIONA UN SEXO \n");
        sexo = Validar(0, 1);
        strcpy(VectReg[encontrado].Sexo, sexos[sexo]);
        break;
    case 5:
        printf("EDITAR EDAD\n");
        VectReg[encontrado].Edad = Validar(18, 65);
        break;
    }
}
system("PAUSE");
}

```

```

        break;
    case 3: // ELIMINAR REGISTRO
        system("CLS");
        if (i == 0)
        {
            printf("INCAPAZ DE ELIMINAR REGISTROS VACIOS\n");
        }
        else
        {
            printf("NUMERO DE EMPLEADO DE REGISTRO A ELIMINAR\n");
            apagar = Validar(1, 2000);
            if (ordenado == 0)
            {
                encontrado = BusquedaTReg(VectReg, i, apagar);
            }
            else
            {
                encontrado = BusquedaBinaria(VectReg, 0, i, apagar);
            }
            if (encontrado == -1)
            {
                printf("NUMERO DE EMPLEADO NO EXISTENTE\n");
            }
            else // Encontrada
            {
                if (VectReg[encontrado].Status == 0)
                {
                    printf("REGISTRO YA ELIMINADO\n");
                }
                else
                {
                    system("CLS");
                    printf("DATOS DEL REGISTRO ENCONTRADO:\n");
                    printf("EMPLEADO EN REGISTRO: %d\n", encontrado);
                    printf("NUMERO DE EMPLEADO: %d\n", VectReg[encontrado].Empleado);
                }
            }
        }
    }
}

```

```

printf("NOMBRE: %s\n", VectReg[encontrado].Nombre);
printf("APELLIDO PATERNO: %s\n", VectReg[encontrado].Paterno);
printf("APELLIDO MATERNO: %s\n", VectReg[encontrado].Materno);
printf("PUESTO: %s\n", VectReg[encontrado].Puesto);
printf("EDAD: %d\n", VectReg[encontrado].Edad);
printf("SEXO: %s\n", VectReg[encontrado].Sexo);
system("pause");
system("cls");
printf("DESEA ELIMINAR EL REGISTRO: 1.SI 0.NO\n");
confirmar = Validar(0, 1);
system("cls");
if (confirmar == 1)
{
    VectReg[encontrado].Status = 0;
    system("CLS");
    printf("REGISTRO ELIMINADO\n");
    system("PAUSE");
}
else
{
    system("CLS");
    printf("REGISTRO NO ELIMINADO\n");
    system("PAUSE");
}
}
}
system("PAUSE");
break;
case 4:
    system("CLS");
    if (i == 0)
    {
        printf("REGISTROS VACIOS\n");
        system("PAUSE");
    }
    else

```

```

{
    printf("NUMERO DE EMPLEADO A BUSCAR\n");
    buscar = Validar(1, 2000);
    if (ordenado == 0)
    {
        encontrado = BusquedaTReg(VectReg, i, buscar);
    }
    else
    {
        encontrado = BusquedaBinaria(VectReg, 0, i, buscar);
    }

    if (encontrado == -1)
    {
        printf("NUMERO DE EMPLEADO NO ENCONTRADA\n");
    }
    else
    {
        if (VectReg[encontrado].Status == 0)
        {
            printf("NUMERO DE EMPLEADO DE ALUMNO DESACTIVADO\n");
        }
        else
        {
            system("CLS");
            printf("DATOS DEL REGISTRO ENCONTRADO:\n");
            printf("EMPLEADO EN REGISTRO: %d\n", encontrado);
            printf("NUMERO DE EMPLEADO: %d\n", VectReg[encontrado].Empleado);
            printf("NOMBRE: %s\n", VectReg[encontrado].Nombre);
            printf("APELLIDO PATERNO: %s\n", VectReg[encontrado].Paterno);
            printf("APELLIDO MATERNO: %s\n", VectReg[encontrado].Materno);
            printf("PUESTO: %s\n", VectReg[encontrado].Puesto);
            printf("EDAD: %d\n", VectReg[encontrado].Edad);
            printf("SEXO: %s\n", VectReg[encontrado].Sexo);
            system("pause");
            system("cls");
        }
    }
}

```



```

    }
    system("PAUSE");
    break;
case 5: // ORDENAR REGISTROS
    system("CLS");
    if (i <= 1)
    {
        if (i == 0)
        {
            printf("INCAPAZ DE ORDENAR UN REGISTRO VACIO\n");
        }
        else
        {
            printf("UN SOLO REGISTRO ORDENADO\n");
        }
    }
    else
    {
        if (ordenado == 0)
        {
            if (i < 500)
            {
                ordenado = OrdenarTReg(VectReg, i);
                printf("REGISTROS ORDENADOS\n");
            }
            else
            {
                Quicksort(VectReg, 0, i - 1);
                ordenado = 1;
                printf("REGISTROS ORDENADOS POR QUICKSORT\n");
            }
        }
        else
        {
            printf("REGISTROS YA ORDENADOS\n");
        }
    }
}

```

```

        system("PAUSE");
        break;
    case 6: // IMPRIMIR REGISTROS
        system("CLS");
        if (i == 0)
        {
            printf("REGISTROS VACIOS\n");
        }
        else
        {
            ImprimirTReg(VectReg, i);
        }
        system("PAUSE");
        break;
    case 7: // GENERAR TXT
        system("CLS");
        if (i == 0)
        {
            printf("REGISTROS VACIOS\n");
        }
        else
        {
            NombreArch(NombreTXT);
            CrearTXT(NombreTXT, VectReg, i);
            creado = 1;
            printf("ARCHIVO CREADO EXITOSAMENTE");
            printf("\n");
        }
        system("PAUSE");
        break;
    case 8: // MOSTRAR ARCHIVO TXT
        NombreArch(Mostrar);
        MostrarTXT(Mostrar);
        system("pause");
        break;
    case 9: // CREAR BINARIO;
        system("CLS");

```

```

system("CLS");
if (creado)
{
    RespaldoBIN(NombreBIN);
    NombreArch(NombreBIN);
    creado = CrearBIN(VectReg, i, NombreBIN);
}
else
{
    NombreArch(NombreBIN);
    creado = CrearBIN(VectReg, i, NombreBIN);
}

system("pause");
break;
case 10: // CARGAR BINARIO
system("CLS");
if (cargado == 0)
{
    NombreArch(NombreBIN);
    cargado = LeerBIN(VectReg, &i, N, NombreBIN);
}
else
{
    printf("ARCHIVO YA CARGADO\n");
}
system("pause");
break;
case 11: // MOSTRAR BORRADOS
system("CLS");
ImprimirApagados(VectReg, i);
system("pause");
break;
case 0:
    if (cargado == 0)

```

```

    {
        if (creado)
        {
            ActualizarTXT(NombreTXT, VectReg, i);
            CrearBorrados(VectReg, i);
        }
        else
        {
            ActualizarTXT(NombreTXT, VectReg, i);
            CrearBorrados(VectReg, i);
        }
        break;
    }
}
} while (op != 0);
}

```

Llenar auto

```
REGISTROS LLENADOS AUTOMATICAMENTE  
Presione una tecla para continuar . . .
```

Editar registro

```
DATOS DEL REGISTRO ENCONTRADO:  
EMPLEADO EN REGISTRO: 2  
NUMERO DE EMPLEADO: 55  
NOMBRE: LUIS  
APELLIDO PATERNO: HERRERA  
APELLIDO MATERNO: LUGO  
PUESTO: GERENTE  
EDAD: 22  
SEXO: HOMBRE  
Presione una tecla para continuar . . .
```

```
EDITAR REGISTRO: 1.SI 0.NO  
1
```

Eliminar registro

```
DESEA ELIMINAR EL REGISTRO: 1.SI 0.NO  
1
```

Buscar

```
DATOS DEL REGISTRO ENCONTRADO:  
EMPLEADO EN REGISTRO: 2  
NUMERO DE EMPLEADO: 55  
NOMBRE: LUIS  
APELLIDO PATERNO: HERRERA  
APELLIDO MATERNO: LUGO  
PUESTO: GERENTE  
EDAD: 22  
SEXO: HOMBRE  
Presione una tecla para continuar . . .
```

Ordenar todo

```
REGISTROS YA ORDENADOS  
Presione una tecla para continuar . . .
```

Imprimir todo

NO.	NO. EMPLEADO	NOMBRE			EDAD	PUESTO	SEXO
0	553	CARLOS	CASTRO	PANTOJA	22	TECNICO	HOMBRE
1	87	MIGUEL	RIOS	ROJAS	22	ANALISTA	HOMBRE
2	55	LUIS	HERRERA	LUGO	22	GERENTE	HOMBRE
3	36	LAURA	VILLANUEVA	MONTOYA	19	EMPLEADO	MUJER
4	961	SERGIO	PEREDES	CASTRO	23	EMPLEADO	HOMBRE
5	929	SARA	MENDOZA	NAVARRO	25	EMPLEADO	MUJER
6	344	FERNANDO	MEDINA	FERNANDEZ	24	GERENTE	HOMBRE
7	1446	EDUARDO	FERNANDEZ	MORALES	27	ASISTENTE	HOMBRE
8	664	FRANCISCO	MEDINA	MONTOYA	27	ASISTENTE	HOMBRE
9	483	GABRIEL	SANCHEZ	CASTRO	18	GERENTE	HOMBRE
10	690	OSCAR	PIZARRO	DURAN	19	TECNICO	HOMBRE

Crear TXT

```
NOMBRE DEL ARCHIVO SIN EXTENSION:
datos
ARCHIVO CREADO EXITOSAMENTE
Presione una tecla para continuar . . .
```

Mostrar archivo

NO.	NO. EMPLEADO	NOMBRE			EDAD	PUESTO	SEXO
40	880	MARIO	ORTEGA	SILVA	27	TECNICO	HOMBRE
41	929	SARA	MENDOZA	NAVARRO	25	EMPLEADO	MUJER
42	961	SERGIO	PEREDES	CASTRO	23	EMPLEADO	HOMBRE
43	980	ALEJANDRO	MEDINA	NAVARRO	27	ASISTENTE	HOMBRE
44	1873	SILVIA	MENDOZA	VARGAS	23	ASISTENTE	MUJER
45	996	GABRIEL	CALDERON	RIOS	26	ANALISTA	HOMBRE
46	1008	FRANCISCO	ROJAS	ZAMORA	27	TECNICO	HOMBRE
47	1022	DANIEL	HARO	GUZMAN	27	GERENTE	HOMBRE
48	1029	VALERIA	VARGAS	SALINAS	20	TECNICO	MUJER
49	1035	ALEJANDRO	SEGURA	REYES	21	ASISTENTE	HOMBRE
50	1058	PABLO	VIDAL	RAMIREZ	24	GERENTE	HOMBRE
51	1144	RODRIGO	CASTRO	MIRANDA	23	GERENTE	HOMBRE

Crear binario

```
NOMBRE DEL ARCHIVO SIN EXTENSION:
hola
Presione una tecla para continuar . . .
```

Cargar binario

```
NOMBRE DEL ARCHIVO SIN EXTENSION:
hola
ARCHIVO CARGADO CON EXITO
Presione una tecla para continuar . . .
```

Eliminados

NO.	NO. EMPLEADO	NOMBRE			EDAD	PUESTO	SEXO
37	71	RODRIGO	GUZMAN	CORDERO	24	EMPLEADO	HOMBRE