Estructura y Representación de Datos

Prof. Tatiana Ilabaca

Primer semestre 2021

Módulo 2
Estructuras de datos estáticas

Arreglos y Estructuras

Objetivos

Lección 4

- Implementar una estructura de arreglos
- Implementar un arreglo de estructuras

Arreglos + Estructuras

Caso 1: Arreglo en una estructura

```
typedef struct Bus
{
    short int nroMaquina;
    char patente[7];
    short int asiento[45];
}bus;

void main()
{
    bus b;
}
```

Un campo de la estructura corresponde a un vector

tatiana.llabaca.upla@gmail.com - 012021

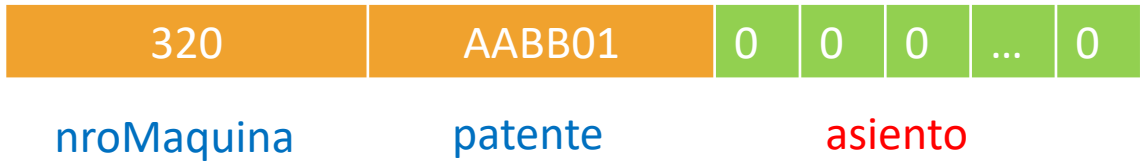
Arreglos + Estructuras

Caso 1: Arreglo en una estructura

- Inicialización de la estructura

```
typedef struct Bus
{
    short int nroMaquina;
    char patente[7];
    short int asiento[45];
}bus;

void main()
{
    bus b={320,"AABB01",{0}};
}
```



tatiana.llabaca.upla@gmail.com - 012021

Arreglos + Estructuras

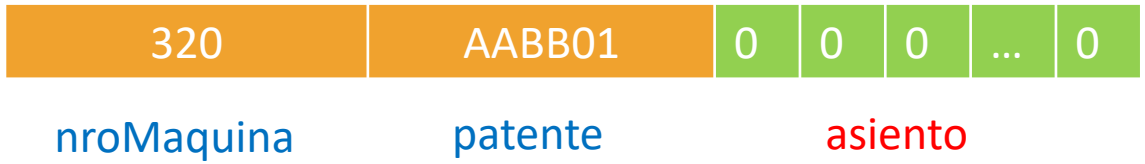
Caso 1: Arreglo en una estructura

- Asignación de valores a la estructura

```
void main()
{
    bus b;
    short int i;

    b.nroMaquina=320;
    strcpy(b.patente,"AABB01");

    for(i=0; i<45; i++)
        b.asiento[i]=0;
}
```



tatiana.llabaca.upla@gmail.com - 012021

Actividad

- Tomando como base el código anterior, agrega nuevas líneas que permitan efectuar la reserva de un asiento.

  Analiza las posibilidades y valida lo que corresponda.

```c
#include <stdio.h>

typedef struct Bus
{
    short int nroMaquina;
    char patente[7];
    short int asiento[45]; //0: Disponible - 1: Reservado
}bus;

void main()
{
    bus b={320,"AABB01",{0}};
}
```


tatiana.llabaca.upla@gmail.com - 012021

Actividad

```
short int nroAsiento;
short int reservado;

do
{
    do
    {
        printf("Ingrese el numero del asiento a reservar: ");
        scanf("%i",&nroAsiento);
    }while(nroAsiento < 1 || nroAsiento > 45);

    if(b.asiento[nroAsiento-1] == 1)
    {
        printf("Asiento ingresado no se encuentra disponible");
        reservado=1;
    }
    else
        reservado=0;

}while(reservado == 1);

b.asiento[nroAsiento-1]=1;
printf("Asiento reservado exitosamente");
```


tatiana.llabaca.upla@gmail.com - 012021

Arreglos + Estructuras

Caso 2: Arreglo de estructuras

```
typedef struct Maquina
{
    short int nroMaquina;
    char patente[7];
    short int agno;
    short int estado;    //1: Operativa - 2: En mantención
}maquina;

void main()
{
    maquina flota[15]; //La flota se compone de 15 buses
}
```


tatiana.llabaca.upla@gmail.com - 012021

Arreglos + Estructuras

Caso 2: Arreglo de estructuras

```
typedef struct Maquina
{
    short int nroMaquina;
    char patente[7];
    short int agno;
    short int estado;
}maquina;

void main()
{
    maquina flota[15];
}
```

Vector flota

0	05	SB3656	2017	1
1	21	FP2452	2016	1
2	12	ED3453	2016	2
3	02	AI8599	2017	1
4	11	HF4113	2015	2
	...			
	...			
	...			
flota[14]	15	CI2551	2017	1

Cada elemento del vector es una estructura

tatiana.llabaca.upla@gmail.com - 012021

Arreglos + Estructuras

Caso 2: Arreglo de estructuras

- Inicialización del arreglo

```
typedef struct Maquina
{
    short int nroMaquina;
    char patente[7];
    short int agno;
    short int estado;   //1: Operativa - 2: En mantención
}maquina;

void main()
{
    maquina flota[15]={{05,"SB3656",2017,1}}; // Primer elemento del vector, una máquina
}
```


tatiana.llabaca.upla@gmail.com - 012021

Arreglos + Estructuras

Caso 2: Arreglo de estructuras

- Despliegue de datos del arreglo

```
void main()
{
    maquina flota[15]={{05,"SB3656",2017,1}};

    printf("%i",flota[0].nroMaquina);
    printf("\n%s",flota[0].patente);
    printf("\n%i",flota[0].agno);
    printf("\n%i",flota[0].estado);
}
```


tatiana.llabaca.upla@gmail.com - 012021

Actividad

- Tomando como base el código anterior, agrega nuevas líneas que permitan cambiar el estado de una máquina.

  Analiza las posibilidades y valida lo que corresponda.

tatiana.llabaca.upla@gmail.com - 012021