

Estructuras de Datos

Prof. Tatiana Ilabaca
Primer semestre 2022



Módulo 2
Estructuras de datos estáticas

Arreglos y Estructuras

Objetivos

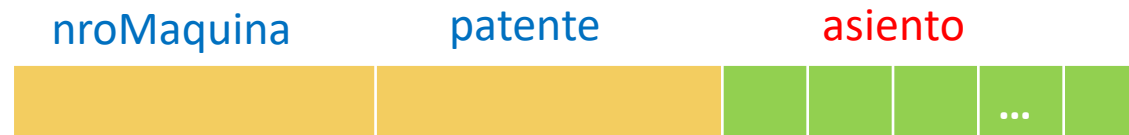
Lección 3

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

Estructuras + Arreglos

Caso 1: Arreglo en una estructura

```
typedef struct Maquina
{
    short int nroMaquina;
    char patente[7];           //Cadena
    short int asiento[45];     //Arreglo (Vector)
}Bus;
```



Un campo de la estructura
corresponde a un vector

```
void main()
{
    Bus bus;  Declaración de la variable bus, cuyo tipo de dato es Bus
}
```

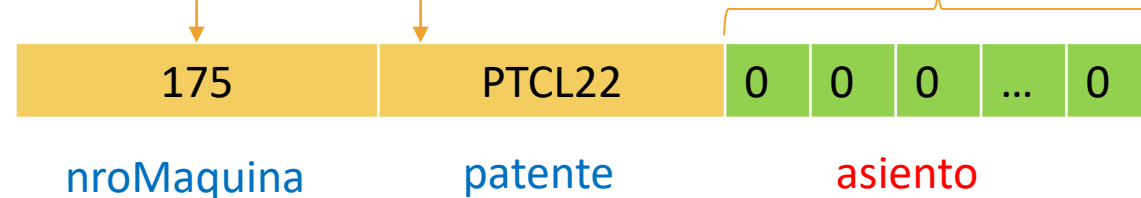
Estructuras + Arreglos

Caso 1: Arreglo en una estructura

- Inicialización de la estructura

```
typedef struct Maquina
{
    short int nroMaquina;
    char patente[7];           //Cadena
    short int asiento[45];    //Arreglo (Vector)
}Bus;
```

```
void main()
{
    Bus bus={175,"PTCL22",{0}};
```



Estructuras + Arreglos

Caso 1: Arreglo en una estructura

- Asignación de valores a la estructura

```
void main()
{
    Bus bus;
    int i;

    bus.nroMaquina=175;
    strcpy(bus.patente,"PTCL22");

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

175	PTCL22	0	0	0	...	0
nroMaquina	patente	asiento				

Actividad 1

- Tomando como base el código anterior, agrega el código que permita efectuar la reserva de un asiento, elegido por el usuario

Analiza las posibilidades y valida lo que corresponda

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

Arreglos + Estructuras

Caso 2: Arreglo de estructuras

```
typedef struct Maquina
{
    short int nroMaquina;
    char patente[7];
    short int agno;
    short int estado;    //1: Operativa - 0: No operativa
}Maquina;
```

```
void main()
{
    Maquina flota[30];    Declaración de la variable flota,
                           cuyo tipo de dato es Maquina
}
```

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[30];
}
```

Vector flota

0	05	SBGT36	2017	1
1	21	FPJH84	2016	1
2	12	EDXZ12	2016	0
3	02	ATPY57	2017	1
4	11	CIFS34	2015	0
	...			
	...			
	...			
flota[30]	15	NLUF95	2017	1

Cada elemento del vector es una estructura

Arreglos + Estructuras

Caso 2: Arreglo de estructuras

- Inicialización del arreglo

```
void main()
{
    Maquina flota[30]={05,"SBGT36",2017,1},{21,"FPJH84",2016,0}};
```

Celda 1

Celda 2

Arreglos + Estructuras

Caso 2: Arreglo de estructuras

- Asignación de datos al arreglo

```
void main()
{
    Maquina flota[30];

    flota[0].nroMaquina=05;
    strcpy(flota[0].patente, "SBGT36");
    flota[0].agno=2017;
    flota[0].estado=1;
}
```

Arreglos + Estructuras

Caso 2: Arreglo de estructuras

- **Despliegue** de datos del arreglo

```
void main()
{
    Maquina flota[30]={05,"SBGT36",2017,1},{21,"FPJH84",2016,0}};

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

Actividad 2

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

Analiza las posibilidades y valida lo que corresponda.