

# HDC

Quiero terminar la sección de programación, así que haré raudamente el tutorial presente. Ya estarán diciendo que es increíble que Roadd haga 2 clases tan corridas xD Es que quiero ser portada de periódicos.

Hoy veremos **estructuras**. Me parece que no es completamente necesario verlas, pero me han ayudado bastante para manejos de servicios y otras chácharas; así que decidí verlos sufrir un poco :D.

**“Estructuras como un edificio? Que tiene que ver con el hacking?”**

No, Manolo. **No**. La estructura es un **tipo de variable**, en la que metemos muchas variables que van relacionadas a esa estructura. Como si tuviésemos una variable **auto** (carro, coche, automóvil) y **dentro de esa variable** tuviésemos los valores de cuanto combustible lleva, si el motor tiene algún problema, cuanto es el kilometraje y cual es la velocidad máxima.



*ojala manejáramos uno de estos xD*

**“Aún así no entiendo como esto podría necesitarse para el hacking”**

Bueno, todavía no es algo que aplicaremos a lo grande pero en resumen nos servirá para crear **datagramas** (como los TCP que vimos anteriormente) o para el manejo de ciertas librerías de sistema.

Para crear una estructura, en código, sería con esta sintaxis:

```
typedef struct {  
    tipo_variable nombre_variable;  
    tipo_variable2 nombre_variable2;  
    ...  
} nombre_variable_estructura;
```

Ejemplo en código:

```

#include <stdio.h>

typedef struct
{
    int numero;
    int numero2;
    char cadena;
} miEstructura;

int main ()
{
    return 0;
}

```

Ahora “**miEstructura**” es un tipo de variable de la misma manera que lo es “**int**”. Es decir que debemos **crear** una variable con ese tipo de variable (podemos crear todas las que queramos). El tema es que tiene más de un valor dentro de cada variable, es una **variable de variables**.

Otra de las cosas que verán es que está declarada antes del **main**. Esto hará que se pueda declarar ese tipo de variable en cualquiera de las funciones que se creen. Si se hubiera declarado dentro del main, sólo funcionaría **dentro** de esa función main.

Declaro la variable y hago unos cambios porque quiero hacerlo más lindo para ustedes.

```

#include <stdio.h>

typedef struct
{
    int patente;
    int caballos_de_fuerza;
    int cantidad_kilometros;
} estructura_auto;

int main ()
{
    estructura_auto miAuto;

    return 0;
}

```

Donde remarqué la línea, creé la variable de tipo **estructura\_auto** (la cual a su vez es una estructura) llamada **“miAuto”**. Para poder darle valores a las variables dentro de la estructura, se hace llamándolas con un punto luego del nombre de la variable.

```
#include <stdio.h>

typedef struct
{
    int patente;
    int caballos_de_fuerza;
    int cantidad_kilometros;
} estructura_auto;

int main ()
{
    estructura_auto miAuto;
    estructura_auto autoVecino;

    miAuto.cantidad_kilometros = 300;
    autoVecino.caballos_de_fuerza = 150;

    return 0;
}
```

Creé otra variable llamada **autoVecino** que me ayudará a que vean como diferenciar una variable de la otra. Entonces cada una de las estructuras creadas tiene su propio conjunto de variables con su propio valor.

**“Y luego de esto debería haber un examen? Creo que voy a morir”**

Vamos que es fácil. Vas a sacarte una buena nota Manolo :).

-----

**Pueden seguirme en Twitter: @RoaddHDC**

**Cualquier cosa pueden mandarme mail a: [r0add@hotmail.com](mailto:r0add@hotmail.com)**

**Para donaciones, pueden hacerlo en bitcoin en la dirección siguiente:  
1HqpPJbbWJ9H2hAZTmpXnVuoLKkP7RFSvw**

**Roadd.**

-----

**Este tutorial puede ser copiado y/o compartido en cualquier medio siempre aclarando que es de mi autoría y de mis propios conocimientos.**