

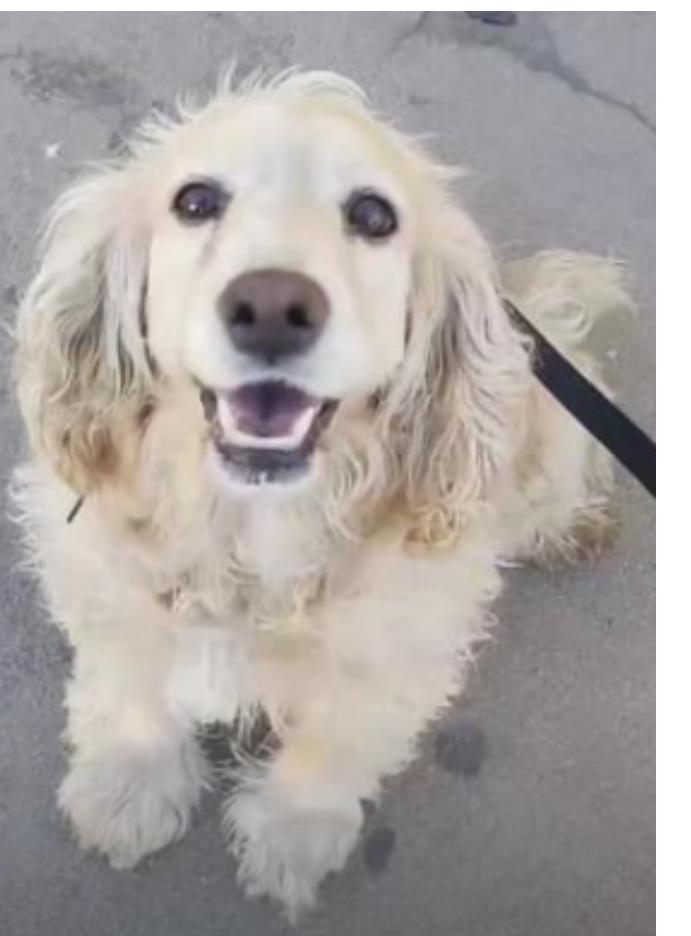
ESTRUCTURAS Y OBJETOS

```
render() {
  return (
    <React.Fragment>
      <div className="py-5">
        <div className="container">
          <Title name="our" title="product">
            <div className="row">
              <ProductConsumer>
                {(value) => {
                  |   |   |   console.log(value)
                  |   |   |
                  |   |   |   }}
                </ProductConsumer>
              </div>
            </div>
          </div>
        </div>
      </React.Fragment>
```

Es un tipo de dato compuesto que permite combinar diferentes tipos de datos relacionados en una sola entidad.

La definición de una estructura se realiza mediante la palabra clave "struct" seguida del nombre de la estructura y las variables miembro entre llaves.

Definir variables para un perrito. Las mismas pueden ser de diferentes tipos.



ESTRUCTURAS

A dark, low-light photograph of a computer keyboard in the foreground, angled towards the viewer. In the background, a computer monitor displays a webpage with various HTML, CSS, and JavaScript code snippets. The monitor also shows some social media links and a navigation menu. The overall composition is moody and technical.

```
#include <iostream>
#include <string>

using namespace std;

// Definición de la estructura Perro
struct Perro {
    string nombre;
    int edad;
    string raza;
};

int main() {
    // Creación de una instancia de la estructura Perro
    Perro perro1;

    // Solicitar al usuario que ingrese los valores del perro
    cout << "Ingrese el nombre del perro: ";
    cin >> perro1.nombre;

    cout << "Ingrese la edad del perro: ";
    cin >> perro1.edad;

    cout << "Ingrese la raza del perro: ";
    cin >> perro1.raza;

    // Mostrar información del perro
    cout << "\nInformacion del perro:" << endl;
    cout << "Nombre: " << perro1.nombre << endl;
    cout << "Edad: " << perro1.edad << endl;
    cout << "Raza: " << perro1.raza << endl;

    return 0;
}
```

ESTRUCTURAS

ESTRUCTURAS EN FORMA DE ARREGLO

```
int main() {  
  
    const int tamanoArreglo = 3;  
    Perro perros[tamanoArreglo];  
  
    // Llenar el arreglo de perros mediante cin  
    for (int i = 0; i < tamanoArreglo; ++i) {  
        cout << "Ingrese el nombre del perro " << i + 1 << ":";  
        cin >> perros[i].nombre;  
  
        cout << "Ingrese la edad del perro " << i + 1 << ":";  
        cin >> perros[i].edad;  
  
        cout << "Ingrese la raza del perro " << i + 1 << ":";  
        cin >> perros[i].raza;  
    }  
  
    // Mostrar la información de los perros almacenados en el arreglo  
    cout << "\nInformacion de los perros:" << endl;  
    for (int i = 0; i < tamanoArreglo; ++i) {  
        cout << "Perro " << i + 1 << endl;  
        cout << "Nombre: " << perros[i].nombre << endl;  
        cout << "Edad: " << perros[i].edad << endl;  
        cout << "Raza: " << perros[i].raza << endl;  
    }  
  
    return 0;  
}
```

VENTAJAS

- Se pueden manejar varios tipos de datos al mismo tiempo.
- Se construyen datos más precisos para cuando se tienen que manejar varios a la vez.
- Retorno de múltiples valores, de manera más precisa.



VENTAJAS DE USAR UNA ESTRUCTURA



EJERCICIO

- Crear una estructura para el objeto "persona". Tomar en cuenta todos los posibles datos que pueda tener.
- Ingresar un arreglo de la estructura "persona", con al menos 5 registros.
- En un menú, digitar las siguientes opciones:
 - Mostrar solo los nombres de las 5 personas.
 - Mostrar el nombre y la edad de las personas de la siguiente manera:
 - Arturo - 25 años.
 - Mostrar las personas en el orden inverso en el que fueron ingresadas.
 - Buscar una persona por su nombre. Por ejemplo, buscar el nombre "Arturo".

EJERCICIO