

**BASES  
DE**

**PROGRAMACIÓN**

EL ARTE DE MANIPULAR DATOS.



# ÍNDICE

```
24 req.user.accessToken;
25 plaidClient.getConnectUser(access_token, {function(err, user) {
26   if (response) {
27     transactions = response.transactions;
28     accounts = response.accounts;
29     User.update({'accessToken': access_token},
30     {
31       $set: {
32         userAccount: accounts,
33         userTransactions: transactions
34       }, {
35         multi: false
36       },
37       function(err, result) {
38         console.log(err);
39         console.log(result);
40       }
41     });
42     res.render('user/account', {title: 'User Account',
43       accounts: accounts,
44       transactions: transactions
45     });
46   } else {
47     User.findOne({'accessToken': access_token}, function (err, user) {
48       transactions = user.userTransactions;
49       accounts = user.userAccount;
50       res.render('user/account', {title: 'User Account',
51         accounts: accounts,
52         transactions: transactions
53       });
54     });
55   }
56 })
```



¿Qué es programar y cómo lo utilizamos?



Variables, tipos, declaraciones y asignaciones



Estructuras de control (if, for, while, switch)

**PROGRAMAR ES  
CÓMO REALIZAR  
ALGO A TRAVÉS DE  
UNA SERIE DE PASOS  
DETALLADOS.**

**PARA ELLO USAMOS  
UN LENGUAJE**

**EN NUESTRO CASO "C"**

# VARIABLES, TIPOS, DECLARACIONES Y ASIGNACIONES

En programación una variable se puede definir sin un valor o con un valor constante.

Cada variable se debe definir con un tipo, los más comunes son: **int** (entero), **float** (decimal), **char** (caracter), **char [ ]** (cadena de caracteres) y **bool** (verdadero o falso). Para hacer uso de una variable, debemos realizar una declaración de la misma.

## ESTRUCTURA DE UNA DECLARACIÓN

The diagram illustrates the structure of two variable declarations. For the first declaration, 'int a = 0;', red arrows point from labels below to the components: 'TIPO' points to 'int', 'NOMBRE' points to 'a', and 'VALOR' points to '0'. For the second declaration, 'int a;', red arrows point from labels below to the components: 'TIPO' points to 'int' and 'NOMBRE' points to 'a'.

```
int a = 0;      int a;
```

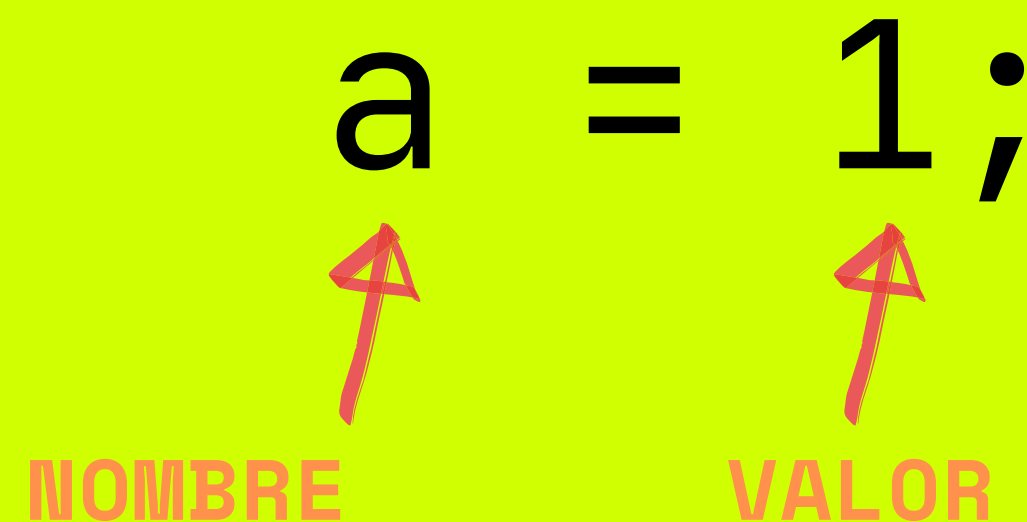
TIPO            NOMBRE            VALOR            TIPO            NOMBRE

## VARIABLES, TIPOS, DECLARACIONES Y ASIGNACIONES

YA QUE HACEMOS USO DE UNA VARIABLE, ESO QUIERE DECIR QUE EN CUALQUIER MOMENTO PODEMOS CAMBIAR SU VALOR, O MEJOR DICHO REALIZAR LA ASIGNACIÓN DE UN NUEVO VALOR.

PARA ESTE CASO NO ES NECESARIO ESCRIBIR EL TIPO, SOLAMENTE DEBEMOS INDICAR UN NOMBRE YA EXISTENTE.

### ESTRUCTURA DE UNA ASIGNACIÓN

  
The diagram shows the assignment statement `a = 1;`. Below the variable `a` is the label `NOMBRE` in red, with a red arrow pointing up to `a`. Below the value `1` is the label `VALOR` in red, with a red arrow pointing up to `1`. The equals sign and semicolon are not labeled.

## ESTRUCTURAS DE CONTROL

**AHORA QUE YA SABEMOS COMO USAR VARIABLES, PODEMOS MANIPULARLAS A TRAVÉS DE CICLOS O FLUJOS PARA UN OBJETIVO ESPECÍFICO. A ESTOS SE LES CONOCEN COMO ESTRUCTURAS DE CONTROL. LOS MÁS COMÚNES SON:**

### **ESTRUCTURA IF**

```
if (a < 2) {  
    a = 3;  
}
```



## ESTRUCTURAS DE CONTROL

### **ESTRUCTURA FOR**

```
for (int a = 0; a < 2; a++) {  
    a = 3;  
}
```



## ESTRUCTURAS DE CONTROL

### **ESTRUCTURA WHILE**

```
while (a = 3){  
    b = 2;  
}
```





## ESTRUCTURAS DE CONTROL

### **ESTRUCTURA SWITCH**

```
switch (a){  
    case 1:  
        b = 2;  
        break;  
}
```



## PRÓXIMA CLASE

# FUNCIONES Y MÉTODOS

Después de hacer uso de las estructuras de control, muchas veces nos es necesario poder organizar estos pasos.

