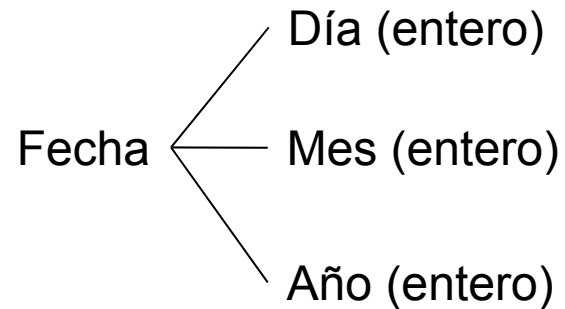
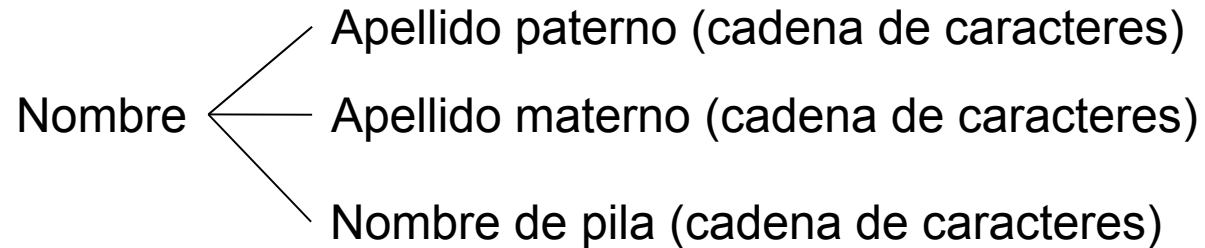


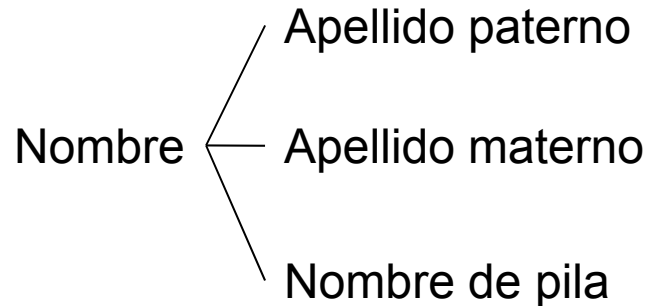
Registros

Registro: Conjunto de elementos que pueden ser de distinto tipo, tratados como una sola entidad y con el mismo identificador (nombre de variable), los elementos, denominados campos, son accesibles a través de su nombre de campo.

Registros



Registros

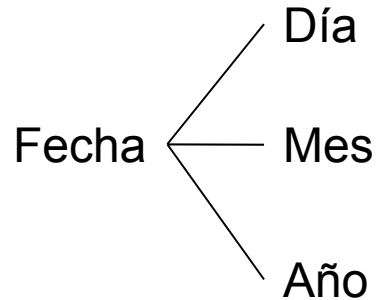


```
struct tipo_nombre {  
    char app[20];  
    char apm[20];  
    char nom[30];  
};
```

```
int main () {  
  
    struct tipo_nombre persona;  
  
    scanf ("%s", persona.app);  
  
    printf ("%s", persona.apm);  
  
    printf ("%c", persona.nom[0]);  
  
}
```

* Cada elemento del registro, accesado a través de su nombre de campo es tratado como una variable independiente

Registros

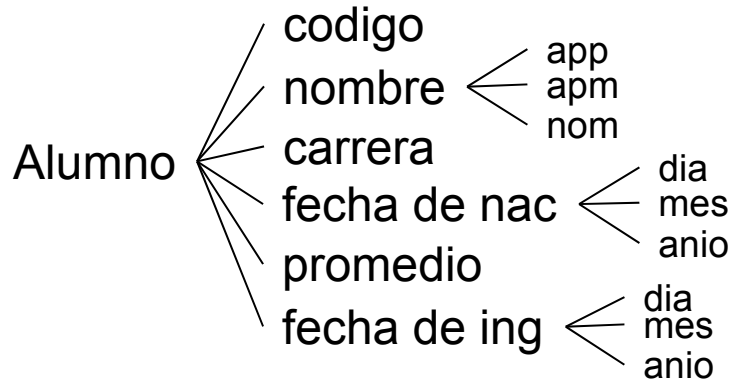


```
struct tipo_fecha {  
    int dia;  
    int mes;  
    int anio;  
};
```

```
int main () {  
  
    struct tipo_fecha nacim;  
  
    scanf ("%d", &nacim.dia);  
  
    printf ("%d", nacim.mes);  
  
    nacim.anio = 1985;  
  
}
```

* Cada elemento del registro, accesado a través de su nombre de campo es tratado como una variable independiente

Registros



```
struct tipo_fecha {  
    int dia, mes, anio;  
};
```

```
struct tipo_nombre {  
    char app[20], apm[20], nom[30];  
};
```

```
struct tipo_alumno {
```

```
    char codigo[10];
```

```
    struct tipo_nombre nombre;
```

```
    char carrera [25];
```

```
    struct tipo_fecha fec_nac;
```

```
    float promedio;
```

```
    struct tipo_fecha fec_ing;
```

```
};
```

Anidación estructural

Capacidad de colocar datos de tipo estructurado dentro de otros datos de tipo estructurado

Registros

```
struct tipo_fecha {  
    int dia, mes, anio;  
};  
  
struct tipo_nombre {  
    char app[20], apm[20], nom[30];  
};  
  
struct tipo_alumno {  
    char codigo[10];  
    struct tipo_nombre nombre;  
    char carrera [25];  
    struct tipo_fecha fec_nac;  
    float promedio;  
    struct tipo_fecha fec_ing;  
};
```

```
int main () {  
  
    struct tipo_alumno a, b;  
  
    scanf ("%s", a.codigo);  
  
    printf ("%s", b.codigo);  
  
    scanf ("%s", a.nombre.app);  
  
    printf ("%s", b.nombre.apm);  
  
    scanf ("%s", a.carrera);  
  
    scanf ("%d", &a.fec_nac.día);  
  
    printf ("%d", b.fec_nac.mes);  
  
    a.fec_ing.anio = 2006;  
  
}
```

Registros

```
struct tipo_fecha {  
    int dia, mes, anio;  
};  
  
struct tipo_nombre {  
    char app[20], apm[20], nom[30];  
};  
  
struct tipo_alumno {  
    char codigo[10];  
    struct tipo_nombre nombre;  
    char carrera [25];  
    struct tipo_fecha fec_nac;  
    float promedio;  
    struct tipo_fecha fec_ing;  
};
```

```
int main () {  
  
    struct tipo_alumno grupo[50];  
  
    scanf ("%s", grupo[13].codigo);  
  
    scanf ("%s", grupo[42].nombre.app);  
  
    printf ("%s", grupo[31].nombre.apm);  
  
    printf ("%s", grupo[31].nombre.nom[0]);  
  
    scanf ("%s", grupo[48].carrera);  
  
    scanf ("%d", &grupo[5].fec_nac.día);  
  
    printf ("%d", grupo[17].fec_nac.mes);  
  
    grupo[27].fec_ing.anio = 2006;  
  
}
```

Registros

```
int main () {  
  
    struct tipo_alumno grupo[50];  
    int i;  
  
    for (i = 0; i < 50; i++) {  
        scanf ("%s", grupo[i].codigo);  
        scanf ("%s", grupo[i].nombre.app);  
        scanf ("%s", grupo[i].nombre.apm);  
        scanf ("%s", grupo[i].carrera);  
        scanf ("%d", &grupo[i].fec_nac.día);  
        scanf ("%d", &grupo[i].fec_nac.mes);  
        .....  
    }  
  
    for (i = 0; i < 50; i++) {  
        printf ("%s", grupo[i].codigo);  
        printf ("%s", grupo[i].nombre.app);  
        printf ("%s", grupo[i].nombre.apm);  
        printf ("%c", grupo[i].nombre.nom[0]);  
        printf ("%s", grupo[i].carrera);  
        printf ("%d", grupo[i].fec_nac.día);  
        printf ("%d", grupo[i].fec_nac.mes);  
        .....  
    }  
}
```