UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

Programación UNSAM
Estructuras

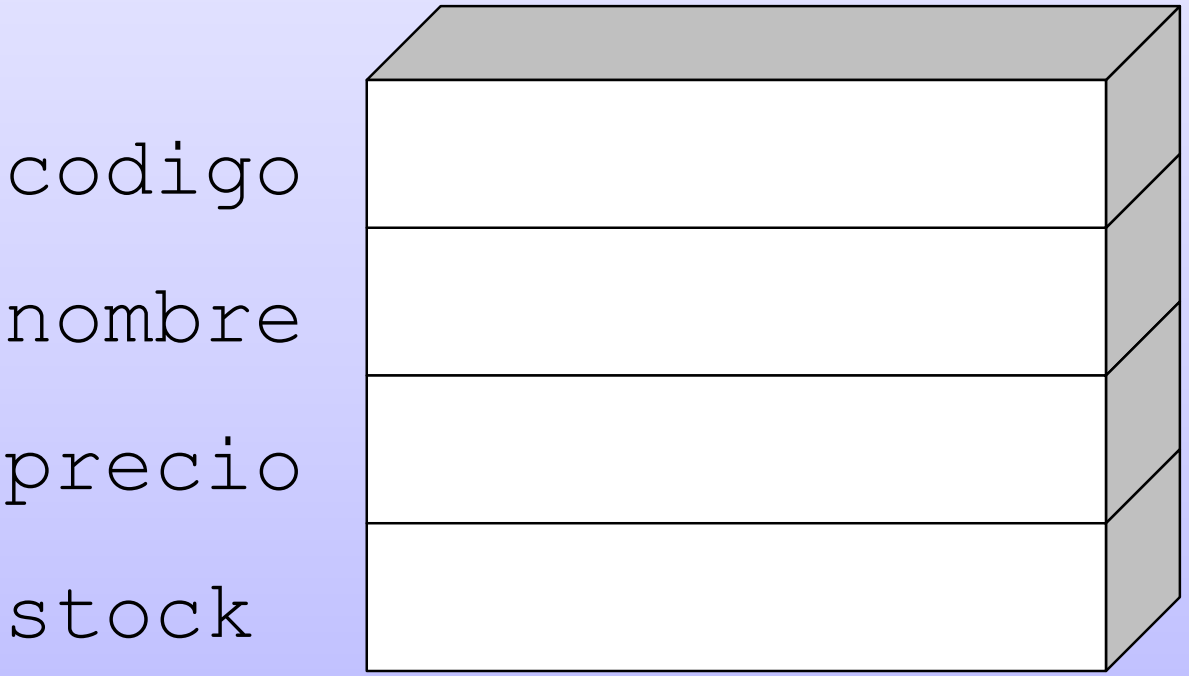
David López

Estructuras (structs)

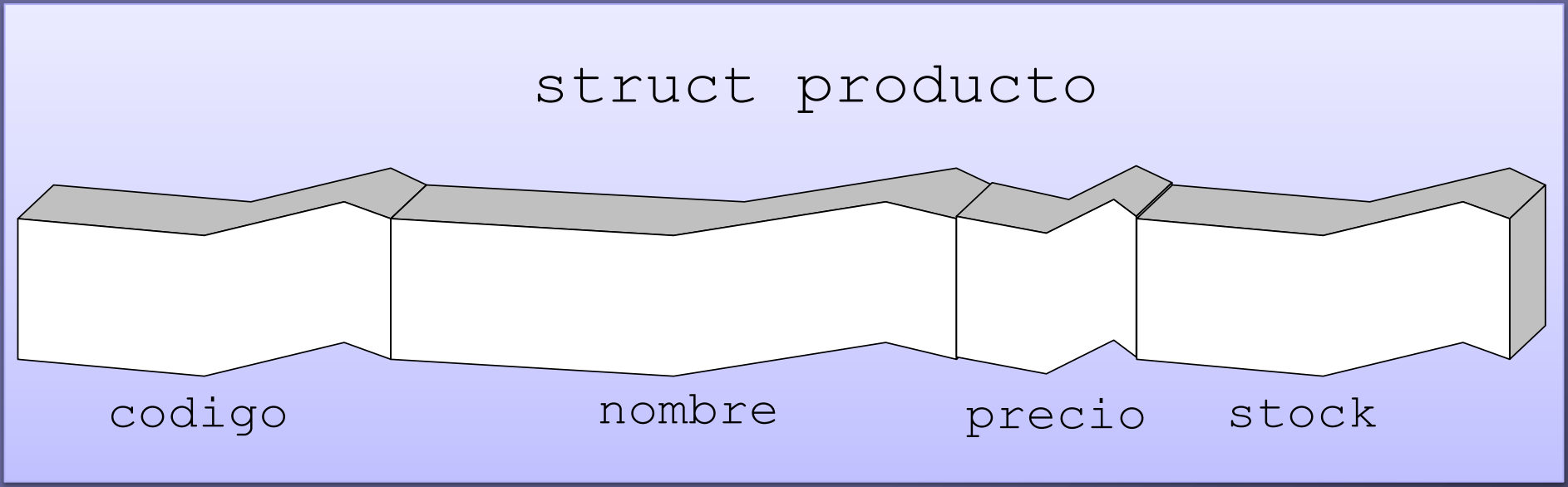
- Sirven para crear un nuevo tipo de dato
  - Compuesto por campos de tipos existentes
- Permiten agrupar varios datos (campos) posiblemente de distintos tipos.
- Permiten a una función retornar varios datos agrupados en una variable.

Representación de una estructura

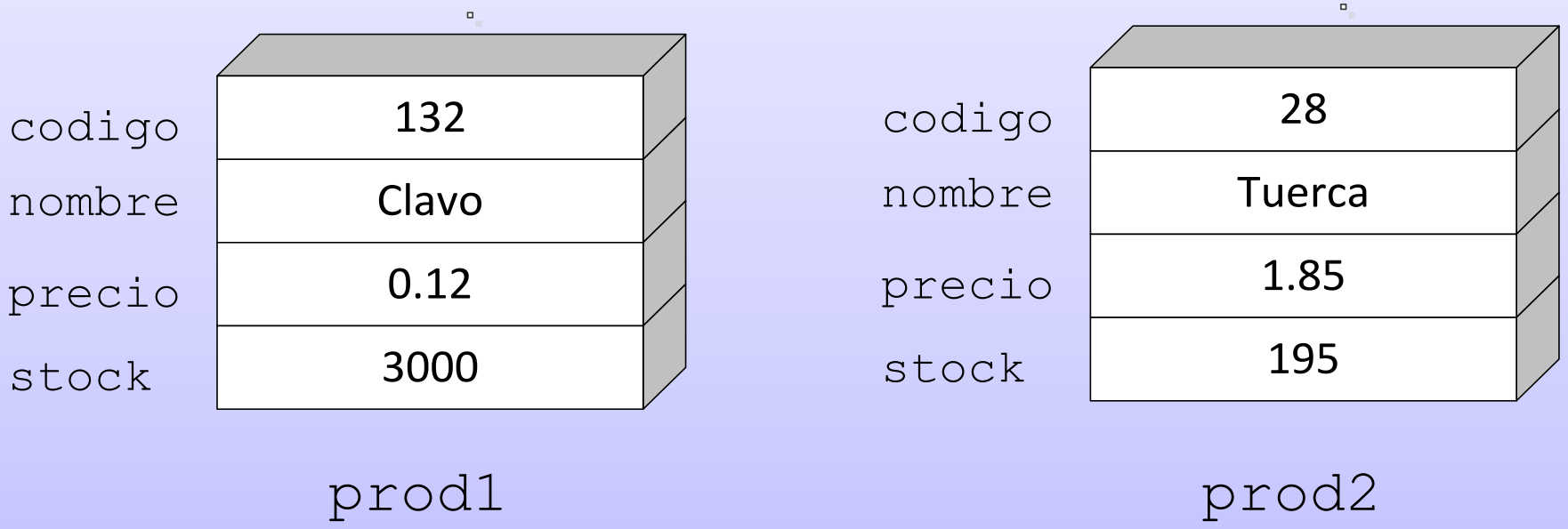
struct producto



Representación de una estructura



Dos variables del mismo tipo de estructura



Sintaxis: Definición del tipo

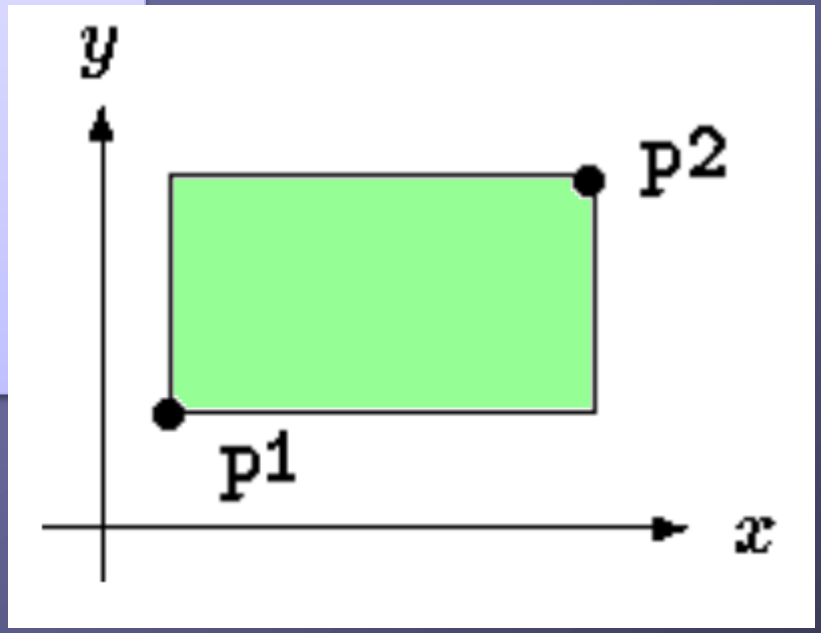
```
struct nombre {
        tipo1 nombre_campo_1;
        tipo2 nombre_campo_2;
        …
        tipoN nombre_campo_N;
};
```

Ejemplo

```
struct producto {
     int codigo;
     char nombre [10];
     float precio;
     long stock;
};
```

Otro ejemplo de uso

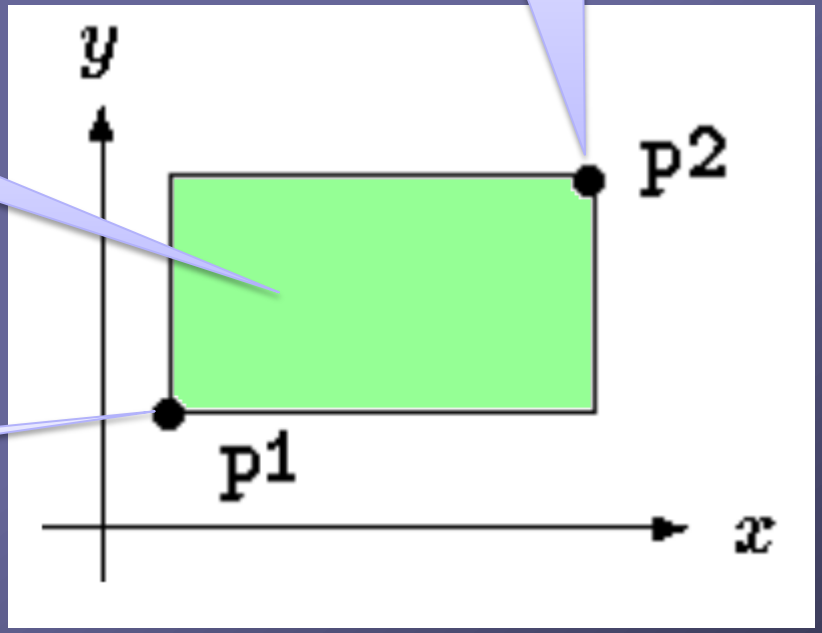
```
/*Estructura con dos coordenadas*/
struct punto {
    int x;
    int y;
};

/*Estructura que contiene dos
estructuras (esquinas)*/
struct rect {
    struct punto p1;
    struct punto p2;
    int color_relleno;
};
```



Otro ejemplo de uso de estructuras (cont.)

```
struct punto p2;
```

```
struct rect r1;
```




```
struct punto p1;
```

Ejemplo: programa completo

```
Ver ejemplo_struct.c
```

Pregunta

- Similitudes, diferencias, ventajas y desventajas respecto de vectores/matrices

Instrucción typedef

- Permite definir un alias a un tipo:
- Sintaxis: `typedef tipo alias;`
- Ejemplo:

```
typedef unsigned int uint;
typedef struct s{
      int campo1;
      float campo2;
} mi_tipo;

//Declaracion de variables
uint a, b;
mi_tipo c, d;
```