Punteros - Conceptos iniciales

¿Qué es un puntero?

Un puntero es una variable que se declara, al igual que los otros tipos de variables. En la declaración se precede del operador * y se le asigna un tipo y un nombre.

Un puntero almacena direcciones de memoria de variables o de espacios en los que se almacenan valores del mismo tipo del puntero.

Un puntero se declara de forma idéntica a la variable a la que va a apuntar, pero se le agrega un * a la izquierda.

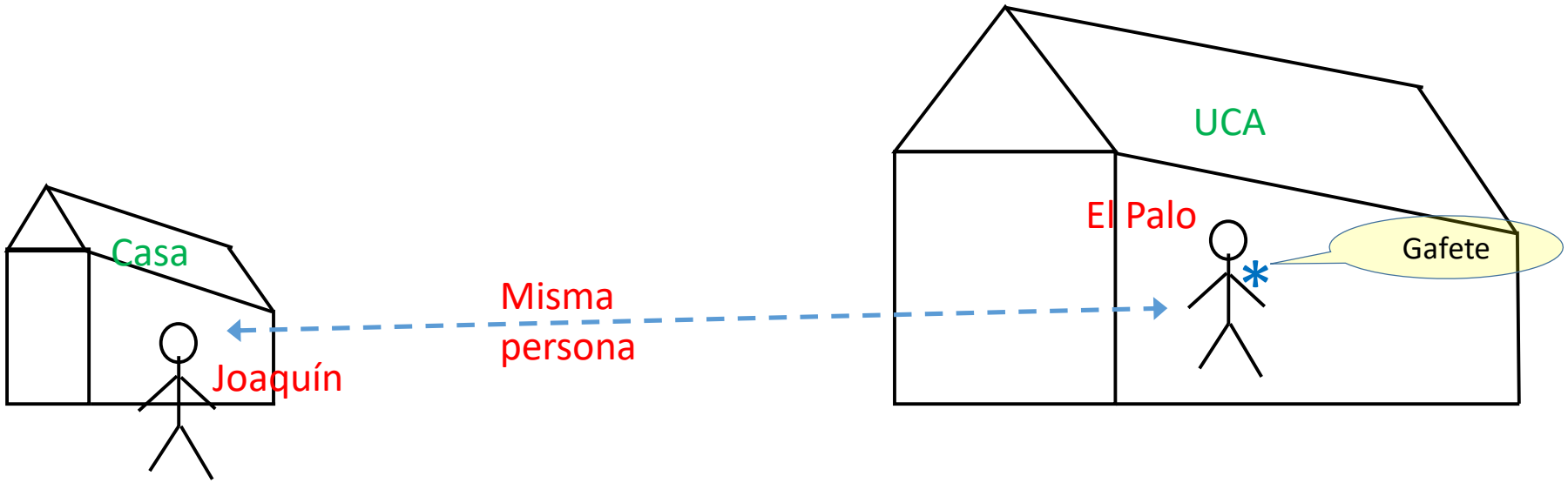
Un puntero no debe, apuntar a un espacio donde hay un valor de un tipo distinto al del puntero. Pero un punteros se puede declarar de tipo void, lo que permite asignarle la dirección de un espacio de un tipo que no se conoce de antemano.

Opreadores & y *

El operador **&** (operador "dirección de") toma la dirección de la variable a la que se le aplica, y retorna esa dirección. Puede utilizarse para asignar la dirección de una variable a un puntero.

El operador ***** (operador "de indirección") toma el valor del espacio de memoria al que está apuntando el puntero.

Punteros - Conceptos iniciales



Con un puntero podemos "representar" a una variable externa dentro de una función, así como el Palo representa a Joaquín en la UCA.

Punteros simples:

Supongamos una variable entera:

```
int a;
```

Supongamos también un puntero a enteros:

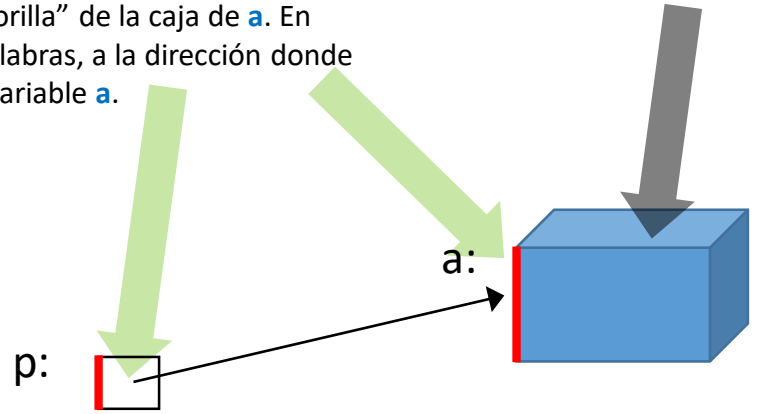
```
int *p;
```

Para hacer que el puntero **p** apunte a la variable **a** hacemos:

```
p = &a;
```

Luego, si en un programa enunciamos **p**, hacemos referencia al contenido de **p**, que no es más que la "orilla" de la caja de **a**. En otras palabras, a la dirección donde está la variable **a**.

Si en un programa enunciamos **a**, hacemos referencia al contenido de **a**, es decir, a lo que esta variable almacena en su interior.



Pero si queremos referirnos al contenido de la variable **a**, en el programa enunciamos ***p**.

Para llegar al valor de **a** a través de **a**, escribimos: **a**

Para llegar al valor de **a** a través de **p**, escribimos: ***p**

Punteros dobles:

Supongamos una variable entera:

```
int a;
```

Supongamos también un puntero a enteros:

```
int *p;
```

Supongamos también un puntero a punteros a enteros:

```
int **q;
```

Para hacer que el puntero **p** apunte a la variable **a** hacemos:

```
p = &a;
```

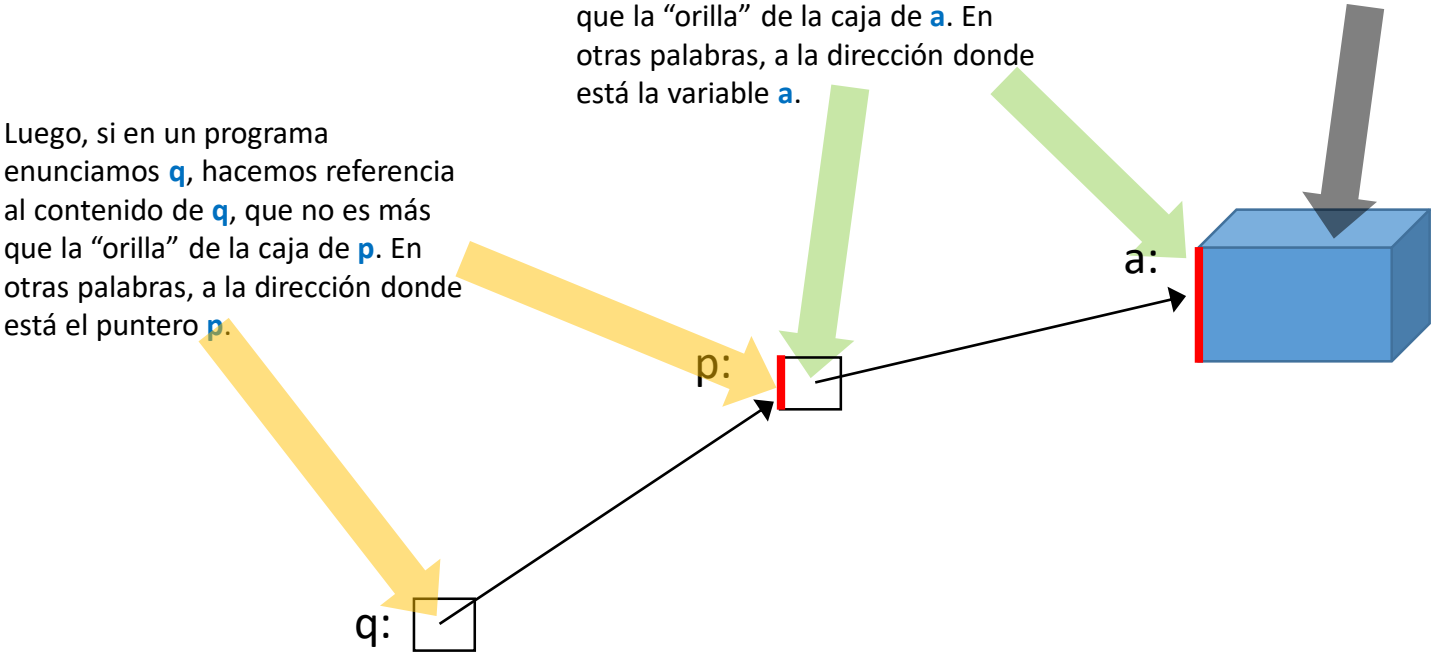
Para hacer que el puntero **q** apunte al puntero **p** hacemos:

```
q = &p;
```

Luego, si en un programa enunciamos **q**, hacemos referencia al contenido de **q**, que no es más que la "orilla" de la caja de **p**. En otras palabras, a la dirección donde está el puntero **p**.

Luego, si en un programa enunciamos **p**, hacemos referencia al contenido de **p**, que no es más que la "orilla" de la caja de **a**. En otras palabras, a la dirección donde está la variable **a**.

Si en un programa enunciamos **a**, hacemos referencia al contenido de **a**, es decir, lo que esta variable almacena en su interior.



Si queremos referirnos al contenido de la variable **p**, en el programa enunciamos ***q**.

Si queremos referirnos al contenido de la variable **a**, en el programa enunciamos ****q**.

Para llegar al valor de **a** a través de **a**, escribimos: **a**

Para llegar al valor de **a** a través de **p**, escribimos: ***p**

Para llegar al valor de **a** a través de **q**, escribimos: ****q**

Punteros triples:

Supongamos una variable entera:

```
int a;
```

Supongamos también un puntero a enteros:

```
int *p;
```

Supongamos también un puntero a punteros a enteros:

```
int **q;
```

Supongamos también un puntero a punteros a punteros a enteros:

```
int ***r;
```

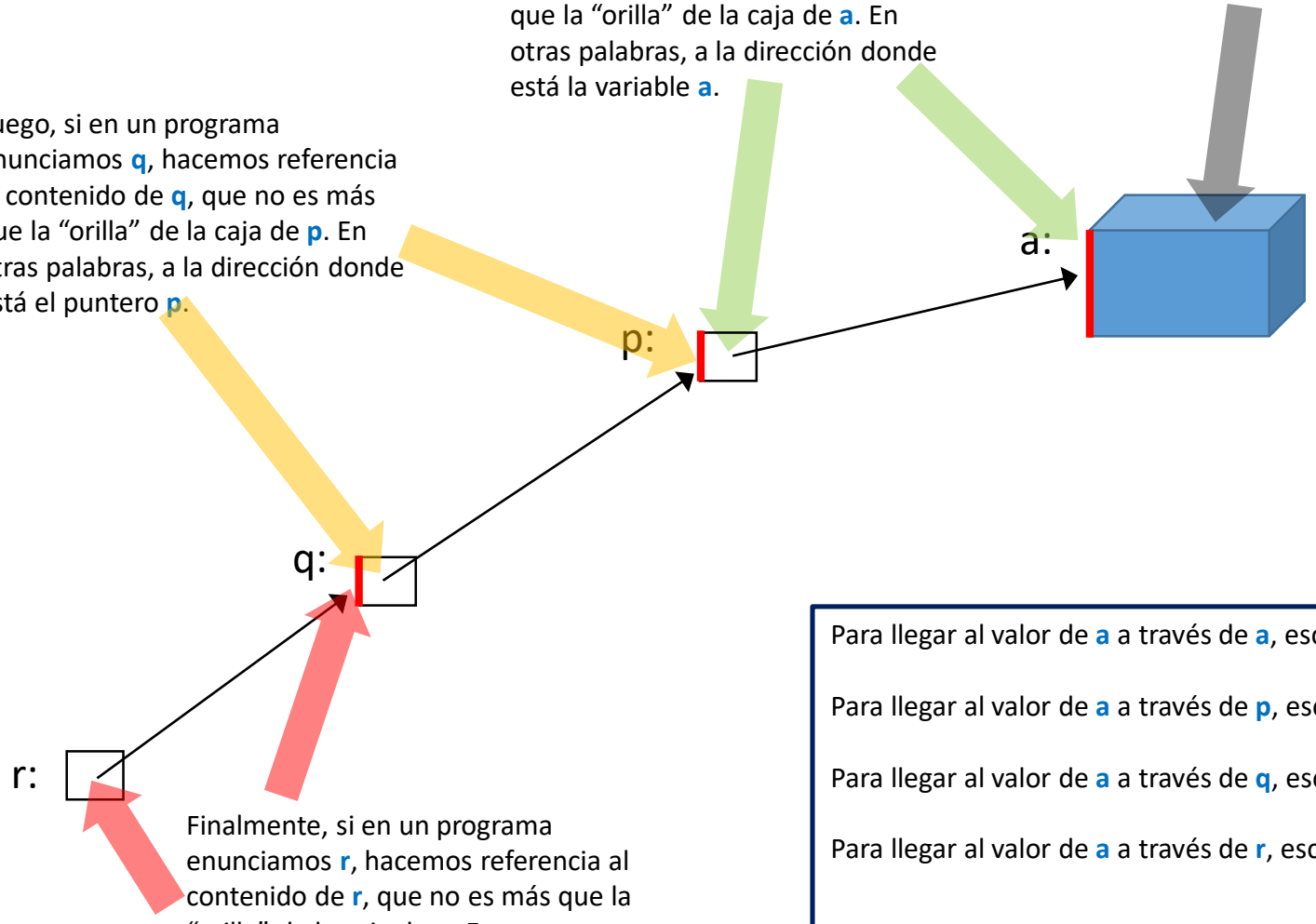
Para hacer que el puntero **r** apunte al puntero **q** hacemos:

```
r= &q;
```

Si en un programa enunciamos **a**, hacemos referencia al contenido de **a**, es decir, lo que la variable almacena en su interior.

Luego, si en un programa enunciamos **p**, hacemos referencia al contenido de **p**, que no es más que la "orilla" de la caja de **a**. En otras palabras, a la dirección donde está la variable **a**.

Luego, si en un programa enunciamos **q**, hacemos referencia al contenido de **q**, que no es más que la "orilla" de la caja de **p**. En otras palabras, a la dirección donde está el puntero **p**.



Finalmente, si en un programa enunciamos **r**, hacemos referencia al contenido de **r**, que no es más que la "orilla" de la caja de **q**. En otras palabras, a la dirección donde está el puntero **q**.

Para llegar al valor de **a** a través de **a**, escribimos: **a**

Para llegar al valor de **a** a través de **p**, escribimos: ***p**

Para llegar al valor de **a** a través de **q**, escribimos: ****q**

Para llegar al valor de **a** a través de **r**, escribimos: *****r**