

```

array_jpush($pila,"cuatro","cinco");
while ($valor = arrayjop($pila) ) {
    echo "valor extraído es $valor<br>";
}
?>

```

El resultado por pantalla muestra que la extracción de los valores se hace en orden inverso del que fueron introducidos:

```

valor extraído es cinco
valor extraído es cuatro
valor extraído es tres
valor extraído es dos
valor extraído es uno

```

## Ordenación de los valores

Finalmente, PHP ofrece una gran variedad de funciones para ordenar *arrays*. Las funciones pueden verse en la tabla siguiente:

**Tabla 7.1.** Funciones de ordenación.

Función	Explicación
asort()	Ordena de forma ascendente el <i>array</i> pasado como argumento. Ordena las parejas índice/valor atendiendo al dato. Es un buen método para los <i>arrays</i> asociativos.
arsort()	Igual que <code>asort()</code> , pero ordena en sentido descendente.
ksort()	Ordena de forma ascendente el <i>array</i> pasado como argumento. Ordena las parejas índice/valor atendiendo esta vez al índice.
krsort()	Igual que <code>ksort()</code> , pero ordena en sentido descendente.
sort()	Ordena de forma ascendente el <i>array</i> pasado como argumento. Se pierde el valor asociativo entre el índice y el valor.
rsort()	Igual, pero en orden descendente.

Puede ver actuar todas las funciones de ordenación en el ejemplo siguiente:

```

<?php
function recorre ($numero) {
    foreach ($numero as $indice => $valor) {
        echo "$indice: $valor<br>";
    }
}

```