Folla 2.8. Funcións de ordenamento de arrays

1. Define o seguinte array:

\$puntos = array("Ana"=>123, "Belén"=>14,"Felipe"=>3,"Moncho"=>245,"Artur"=>10);

Executa a función **sort()** sobre o array e móstrao cun foreach, para ver o array ordenado. Verás que esta funcións non mantén a asociación das claves cos valores (elimina as claves).

2. Fai unha páxina titulada *Ordenamento de arrays en PHP*, suliñado. Fai agora unha táboa para mostrar o que fan as funcións de ordenamento de cadeas, facendo previamente <u>cada vez unha copia do array e operando sobre esta copia</u> (para non modificar os arrays iniciais): sort(), rsort(), asort(), arsort(), ksort(), shuffle(), array_reverse(). A táboa debe ser como esta (emprega print_r na última columna):

Nome Función	Explicación do que fai	Exemplo	Mostra por pantalla
asort()	Ordena un array de menor a maior mantendo a asociación do elemento co seu índice	asort(\$copia)	Felipe=>3 Artur=>10 Belén=>14 Ana=>123 Moncho=>245

3. Fai unha páxina que permita xestionar os libros dunha biblioteca. Deberá ser unha páxina similar á que se mostra na figura (engade unha imaxe antes do título BIBLIOTECA):



cos libros que mostran na figura. Este será resultado de premer "Ver listado completo da biblioteca".

Se se introduce algo na caixa de texto do formulario deberá buscar todos os exemplares que teñan esa cadea de texto tanto no título como no autor e a editorial. Por exemplo, se buscamos a cadea "de" deberá sair:



Emprega funcións de ordenamento de arrays para ordenar polo título, e funcións de tratamento de arrays para buscar o introducido na caixa de texto nos campos TITULO, AUTOR ou EDITORIAL.