

macenar, en un momento dado, y recuperar justo en el mismo estado en el que se encontraba cuando lo guardamos. Esta técnica se llama señalización y permite almacenar y recuperar el conjunto de *bits* que forman un objeto.

PHP 5 ofrece dos funciones, `serialize()` y `unserialize()`, que toman como parámetro un objeto y devuelven una cadena de caracteres. Los objetos serializados los podemos almacenar en base de datos, ficheros, etcétera.

```
$luis = new Apellido("Luis Miguel");
echo $luis->getNombre() . "<br>";
echo $luis->getApellidos() . ",-";
$manuel = serialize($luis);
$pedro = unserialize($manuel);
echo $pedro->getNombre();
```

El código anterior crea el objeto `$luis` y utiliza los métodos propios de la clase y los heredados. Después, utilizamos la función `serialize()` para grabar el estado del objeto como un conjunto de caracteres en la variable `$manuel`. La variable `$pedro` contiene el resultado de aplicar `unserialize()` a `$manuel`.

De esta forma, conseguimos recuperar el objeto `$luis` con los datos que ya habíamos almacenado.

Funciones de manejo de clases

Las funciones siguientes son útiles para recuperar información sobre la herencia de las clases, sobre métodos o variables miembro o llamadas a las funciones.

Tabla 9.1. Funciones de clases.

Función	Descripción
<code>get_class()</code>	Devuelve el nombre de la clase que implementa el objeto pasado como parámetro.
<code>get_parent_class()</code>	Devuelve el nombre de la clase padre.
<code>class_exists()</code>	Si el parámetro es el nombre de una clase, devuelve un <i>true</i> .
<code>get_declared_classes()</code>	Devuelve un <i>array</i> con el nombre de todas las clases definidas en el código actual.

Función	Descripción
<code>is_subclass_of ()</code>	Toma dos parámetros. Si el primer parámetro es una subclase del segundo parámetro la función devuelve un <i>true</i> .
<code>is_a()get_class_vars()</code>	Devuelve un <i>array</i> con parejas variable / valor de una clase, que tengan valores por defecto.
<code>get_object_vars ()</code>	Devuelve un <i>array</i> con parejas variable / valor de un objeto, que tengan valores por defecto.
<code>method_exists ()</code>	Toma dos parámetros. Si el objeto pasado tiene un método con el nombre del segundo parámetro, la función devuelve <i>true</i> .
<code>get_class_method ()</code>	Devuelve un <i>array</i> de métodos definidos en una clase.

Puede ver el funcionamiento de la función `get_class ()` en el ejemplo siguiente y el resultado en la figura 9.5.

```
$luis = new Nombre;  
echo $luis->NombreDefecto();  
echo 'La variable $luis es un objeto del tipo: '  
get_class($luis);
```