

La programación orientada a objetos, además de permitir encapsular los datos relacionados y las operaciones que trabajan sobre ellos, permite la construcción de ciertos esquemas complejos de uso común en complejos fuertemente basados en este paradigma.

Estos esquemas se denominan **patrones de diseño**, y aunque su conocimiento y utilización es un concepto avanzado, intentaremos dar aquí unos ejemplos de alguno de ellos.

Singleton: Este patrón implica que al crear un nuevo objeto de una clase que sigue el patrón **singleton**

```
class Singleton {
    private static $instancia = null;
    private $contador;
    private function __construct() {
        echo "Creación " . __CLASS__ . "<br>";
        $this->contador=0;
    }
    public static function getInstance() {
        if ( self::$instancia == null ) {
            self::$instancia = new self;
        }
        return self::$instancia;
    }
    public function incrementar() { return ++$this->contador; }
    public function disminuir() { return --$this->contador; }
}
```

Factoría: Es una clase que genera diferentes clases de objetos que cumplen todos el mismo interfaz.

```
class HTMLTagFactory {
    public function create($type) {
        switch($type) {
            case 'inputText':
                return new HTMLInputText();
            break;
            case 'checkbox':
                return new HTMLInputCheckbox();
            break;
            case 'select':
                return new HTMLSelect();
            break;
        }
    }
}
```

Iterador: Un **iterador** es un interfaz que se puede implementar con una clase para que recorra todos los elementos de una colección, potencialmente realizando una operación sobre ellos.

```
abstract class Iterator implements Traversable {
    abstract public function current();
    abstract public function key();
    abstract public function next();
    ...
}
```

```
abstract public function rewind();  
abstract public function valid();  
}
```

PHP ya ofrece implementados muchos iteradores para muchos tipos distintos de colecciones de elementos, ver referencia:<http://php.net/manual/es/spl.iterators.php>